A Survey of Adversarial Defences and Robustness in NLP

SHREYA GOYAL, Robert Bosch Centre for Data Science and AI, Indian Institute of Technology Madras, India
SUMANTH DODDAPANENI, Robert Bosch Centre for Data Science and AI, Indian Institute of Technology Madras, India
MITESH M. KHAPRA, Robert Bosch Centre for Data Science and AI, Indian Institute of Technology Madras, India
BALARAMAN RAVINDRAN, Robert Bosch Centre for Data Science and AI, Indian Institute of Technology Madras, India

In the past few years, it has become increasingly evident that deep neural networks are not resilient enough to withstand adversarial perturbations in input data, leaving them vulnerable to attack. Various authors have proposed strong adversarial attacks for computer vision and Natural Language Processing (NLP) tasks. As a response, many defense mechanisms have also been proposed to prevent these networks from failing. The significance of defending neural networks against adversarial attacks lies in ensuring that the model's predictions remain unchanged even if the input data is perturbed. Several methods for adversarial defense in NLP have been proposed, catering to different NLP tasks such as text classification, named entity recognition, and natural language inference. Some of these methods not only defend neural networks against adversarial attacks but also act as a regularization mechanism during training, saving the model from overfitting. This survey aims to review the various methods proposed for adversarial defenses in NLP over the past few years by introducing a novel taxonomy. The survey also highlights the fragility of advanced deep neural networks in NLP and the challenges involved in defending them.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: Adversarial attacks, Adversarial defenses, Perturbations, NLP

ACM Reference Format:
Shreya Goyal, Sumanth Doddapaneni, Mitesh M. Khapra, and Balaraman Ravindran . 2023. A Survey of Adversarial Defences and Robustness in NLP. 1, 1 (April 2023), 43 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Recently, there have been significant advancements in the field of Natural Lanaguage Processing (NLP) using deep learning algorithms. In fact, the proposed solutions for NLP have already surpassed human accuracy in some cases [32, 60, 69]. By learning from vast amounts of available data, deep learning has revolutionized the field by providing a representation of language that can be utilized for a range of tasks. NLP involves the manipulation and processing of human language, and deep neural networks have enabled NLP models to learn how to represent language and solve

Authors' addresses: Shreya Goyal, Robert Bosch Centre for Data Science and AI, Indian Institute of Technology Madras, Bhupat and Jyoti Mehta School of Biosciences, Chennai, Tamil Nadu, India, 600036; Sumanth Doddapaneni, Robert Bosch Centre for Data Science and AI, Indian Institute of Technology Madras, Bhupat and Jyoti Mehta School of Biosciences, Chennai, Tamil Nadu, India, 600036; Mitesh M. Khapra, Robert Bosch Centre for Data Science and AI, Indian Institute of Technology Madras, Bhupat and Jyoti Mehta School of Biosciences, Chennai, Tamil Nadu, India, 600036; Balaraman Ravindran, Robert Bosch Centre for Data Science and AI, Indian Institute of Technology Madras, Bhupat and Jyoti Mehta School of Biosciences, Chennai, Tamil Nadu, India, 600036.

Manuscript submitted to ACM

tasks such as text classification, natural language inferencing, sentiment analysis, machine translation, named entity recognition, malware detection, reading comprehension, textual entailment with remarkable accuracy [60, 96, 128]. A typical NLP pipeline using deep neural networks learns word representation in text and contextual details in sentences, and this type of language modeling can be utilized for tasks such as sentence classification, translation, and question answering, using Convolutional Neural Network (CNN) or Recurrent Neural Networks (RNN) based learning models. Finding feature representations for natural language text is a crucial part of this pipeline, and various methods have been proposed, including hand-crafted features and auto-encoded features using recent deep neural networks [68, 96, 128].

Despite the significant progress made, deep neural networks still suffer from a lack of interpretability and operate as a "black box" [16]. Their high performance remains inexplicable, and there is limited understanding of how they function [16, 113]. Although they can achieve exceptional accuracy and human-like performance, these networks are vulnerable to attacks and are highly sensitive to even the slightest perturbations in inputs, causing them to fail [2, 32]. Recently, there has been a growing number of proposed adversarial attacks for deep neural networks in computer vision and NLP, raising concerns about the robustness of these high-performing models [159, 167]. Adversarial attacks can pose a significant security threat to applications such as malware and spam detection, as well as biometrics. In NLP, these attacks can take various forms, including substitution, insertion, deletion, and swapping of words/characters in a sentence or finding a neighboring word embedding to introduce perturbations in the input [167].

There are two main types of adversarial attacks, black box and white box, based on the attacker's access to the model's parameters [167]. These attacks can be further categorized based on their design granularity, including character level, word level, sentence level, and multi-level attacks [45, 137, 167]. Adversaries are generated by perturbing the input text using techniques such as insertion, deletion, flipping, swapping of characters or words, or paraphrasing the sentence in a way that preserves its original meaning but changes the wording. In white box attacks, the attacker has access to the model's parameters and modifies the word embeddings of input text using gradient-based schemes. In contrast, black box attacks do not have access to the model's parameters and generate a replica of the model by repeatedly querying the input and output. Once the parameters are acquired, they train a substitute model with perturbed data and attack it [45, 103, 137, 159, 167]. Generating perturbations in textual data is more challenging than in images due to the discrete nature of the data [167]. The quality of adversarial examples generated for text data is determined by two factors, namely, the naturalness of the adversarial examples and the efficiency to generate these examples [71]. Some researchers have succeeded in detecting perturbations in text using simpler techniques like spell check and adversarial training [102], while others who used word-level attacks failed to efficiently generate adversarial examples due to the high-dimensional search space [160]. Thus, efficiently generating adversarial attacks in NLP poses unique challenges. Despite the difficulty, stronger and imperceptible adversarial attacks have been proposed, which pose a significant threat to the security of deep neural networks [5, 13]. Consequently, several defense mechanisms have been proposed in recent years to counter adversarial attacks in NLP, and the considerable amount of work in adversarial defenses has provided good competition to the novel adversarial attack algorithms, substantially improving the robustness of existing deep learning models.

Adversarial defense strategies in NLP can be broadly classified into three categories: adversarial training-based, perturbation control-based, and certification-based methods. The majority of the work in this field employs an adversarial training approach and techniques are further subdivided based on the generation of adversarial instances or noise in the defense pipeline. These methods include data augmentation-based adversarial training, adversarial training as a regularization technique, Generative Adversarial Network (GAN)-based adversarial training, Virtual Adversarial Training (VAT), and Human-In-The-Loop (HITL) approaches. Perturbation control-based methods are also categorized

into perturbation identification and correction and perturbation direction control. Certification-based techniques fall under the third category and provide certificates of robustness against adversarial attacks. A few methods do not fit into any of the aforementioned categories and are classified as miscellaneous. In the subsequent section, the objectives of this survey paper are emphasized, distinguishing it from previous surveys.

1.1 Goals of this survey paper

In this article, we reviewed numerous methods of adversarial defenses in NLP, proposed in recent years. The key goals of this survey are as listed below:

- Providing a comprehensive review of adversarial defense schemes in NLP by covering schemes for different NLP tasks and bringing the attention of the community to this emerging area.
- Proposing novel taxonomy for adversarial defense methods in NLP for various tasks.
- Accentuating the importance of defense methods for adversarial attacks and as a regularization scheme in deep neural networks.
- Paving the path for future work in this area by highlighting the open issues.

Numerous survey papers have been published in the past discussing adversarial attacks on deep neural networks in both computer vision and NLP. For example, in [148], the authors conducted a comprehensive survey of adversarial attacks on deep neural networks for images, text, and graphs, proposing a novel taxonomy to categorize a wide range of methods. Similarly, [2] proposed a taxonomy for different adversarial attacks and defenses on various computer vision algorithms, including image classification, image segmentation, object detection, robotic vision, and visual question answering. In addition, [97] presented a brief survey on general adversarial attack methods in deep learning, while [62] briefly reviewed attack and defense methods in images and text data. Furthermore, [17] and [11] discussed adversarial attacks and defenses for various computer vision algorithms.

In contrast to previous research, [164] proposed a novel taxonomy for universal adversarial attacks, which includes universal perturbations for image classifiers, and briefly discussed attacks on text and audio classification models. While the previously discussed survey papers focused primarily on adversarial attacks on images and briefly discussed attack algorithms in NLP, [45, 103, 137, 167] extensively reviewed adversarial attack algorithms for various NLP tasks while briefly discussing some defense methods. However, the importance of adversarial defense algorithms is self-evident, given the large amount of work in this area in recent years. Therefore, this survey paper aims to address this gap and is different from previous survey papers in this area by focusing exclusively on adversarial defense methods in NLP. This paper proposes a detailed and novel taxonomy for adversarial defense mechanisms in NLP, emphasizes the importance of defense methods, and discusses open issues in this area while presenting future work to the community.

In this paper, Section 2 discusses adversarial attacks in deep learning and categorizes adversarial attacks in NLP. Section 3 presents a novel taxonomy for adversarial defense methods in NLP. Section 4 provides a detailed description of adversarial training-based defenses in NLP, along with sub-categories. Section 5 discusses perturbation control-based adversarial defense methods. Section 6 outlines certification-based adversarial defenses. Section 7 describes various other adversarial defenses that do not fit into the previous categorization. Section 8 discusses different metrics used to evaluate defense mechanisms. Section 9 describes the datasets and frameworks proposed for training and evaluating adversarial defense methods. Section 10 provides suggestions for future research in adversarial defenses for NLP, and finally, Section 11 concludes the paper.

2 A GENERAL OVERVIEW OF ADVERSARIAL ATTACKS

An adversarial attack is a deliberate attempt to corrupt a deep neural network's functionality by introducing distorted inputs that cause the model to fail. These perturbations are designed to be subtle enough to evade human detection but effective enough to deceive a neural network. For instance, image classification models have been subjected to experiments with various input perturbations, including the addition of noise, the adjustment of pixels, the use of patches, the addition of watermarks, and so on, which can go unnoticed by humans. In contrast, adversarial attacks in NLP involve multiple proposed perturbations at the character, word, or sentence level through deletion, insertion, swapping, flipping, use of synonyms, concatenation with characters or words, insertion of numeric or alphanumeric characters, etc. However, it is more challenging to generate adversarial perturbations for text data than image data because altering a character or word in a sentence is more perceptible to humans. Moreover, creating imperceptible adversarial attacks is difficult in NLP since perturbations in textual data could result in less natural input data [71]. Adversarial attacks are classified into two categories based on motivation: targeted attacks and non-targeted attacks. Targeted attacks aim to misclassify inputs to a specific class, while non-targeted attacks aim to push the classifier boundary to cause the model to misclassify inputs. Based on access to the model's parameters, adversarial attacks are classified as white-box and black-box attacks. In this section, we briefly review the state-of-the-art adversarial attacks for NLP tasks algorithms.

2.1 Adversarial attacks in deep learning

The goal of an adversarial attack is to generate such perturbations for input $\mathbf{x}$ belonging to class $\mathbf{C_1}$, such that, $\mathbf{x}$ is wrongly classified to class $\mathbf{C_2}$ with a high confidence value. For a multi-class classification algorithm, for k input classes $i = C_1, C_2, ..., C_k$, the perturbed input is $\mathbf{x}^{'}$ and $\mathbf{f_i}$ is the discriminant function which defines the classification boundaries, where $\mathbf{x}$ belongs to class $\mathbf{C_i}$, and $\mathbf{C_{target}}$ is the target class after the attack, then:

$$f_{target}(\mathbf{x}') > f_i(\mathbf{x}') \tag{1}$$

Hence, adversarial attacks can be formally defined as an optimization problem for $\mathbf{x}$ as:

$$\begin{aligned} &\underset{x}{\text{minimize}}(||x' - x||) \\ &subject\ to \underset{i \neq target}{\max}\{f_i(\mathbf{x}')\} - f_{target}(\mathbf{x}') \leq 0 \end{aligned} \tag{2}$$

The inequality defined in 1 represents the goal of any adversarial attack which pushes the perturbed input $\mathbf{x}'$ to a desired target class, rather than its actual class. Hence Equation 2 defines the adversarial attack as an optimization problem, where the goal is to minimize the perturbation magnitude to make the perturbations less perceptible and make sure it gets classified to the target class, $\mathbf{C_{target}}$ [87, 97, 108, 148].

2.2 Adversarial attacks in NLP

In the past few years, numerous methods for adversarial attacks have been introduced, which are specifically designed for NLP tasks. It is important to note that adversarial examples in computer vision cannot be directly applied to text as they are fundamentally different. Therefore, several attack methods that modify the text data while maintaining imperceptibility to humans have been proposed in literature. Typically, these methods alter the text data at the word, character, or sentence levels. The following section presents some of these attack methods in NLP.

Character level adversarial attacks Character-level attacks perturb the input sequences at a character level. These operations include insertion, deletion, and swapping of characters in a given input sequence. Despite the fact, these attacks are quite effective, they can easily be detected with a spell-checker mechanism. One of the techniques used in character-level attacks is adding natural and synthetic noise to the inputs [8]. For natural noise authors collected natural spelling mistakes and used them to replace words in inputs. For synthetic noise, they swap or randomized characters (except the peripheral) and replace a character with its neighboring character on the keyboard. Adding punctuation marks, and increasing or removing the space between characters is another technique to add synthetic noise in the text inputs. For example, in DeepWordBug [33], in black-box setting they use a two-step process as they don't have access to the gradients, parameters, or structure of the model. The first step involves finding the most important words in the sentence which would be the target words to perturb. In the second stage, perturbations are added to these select words by the above-mentioned operations. Edit distance is further used in order to keep track of the readability of the generated sentences. Another example proposed is, TextBugger [74] in both black-box and white-box settings where the white-box attack is a two-step process. The first process involves finding the most important word with the help of the jacobian matrix ($\mathcal{J}_{\mathcal{F}}(x)$) defined as $\mathcal{J}_{\mathcal{F}}(x) = \frac{\partial \mathcal{F}(x)}{\partial x} = \left[\frac{\partial \mathcal{F}_i(x)}{\partial x_i}\right]_{i \in 1...N, j \in 1...K}$, where x_i is the i^{th} word of the input text, N is the total number of words in the input text, and $\mathcal{F}$ is the classifier, and later use 5 different options to add bugs. These 5 include insert, delete, swap, substitution with visually similar words, and substitution with a semantically similar words. In the black-box setting they propose a 3-step process where first the most important sentence is identified, then they find the important words to generate 5 bugs and select the optimal from that. The best adversary is chosen based on how optimal they are for reducing accuracy. Along the same line, [42] has shown that just by adding extra "." (period), spaces between words, "Perspective" API created by Google gave lesser toxicity scores for the words perturbed in this fashion.

Word level adversarial attacks Word level attacks perturb the whole word instead of a few characters. Common operations include insertion, deletion and replacement. Word level attacks can be classified into Gradient-based and Importance based and replacement-based strategies on the basis of the perturbation schemes used:

- In gradient-based methods, the gradient is monitored for every input perturbation. Whenever the probability of classification is reversed that particular perturbation is chosen. This is inspired by the Fast Gradient Sign Method (FGSM) [36] used for adversarial attacks in computer vision models. If the classification probability changes the class then the perturbation is considered effective. Another way of using gradient based method is to find the important words using FGSM and then employ insertion, deletion and replacement strategies on top of them [118]. [82] used a similar approach where they created adversaries by backpropagating for the cost gradients.
- In importance-based methods it is believed that words with the highest or lowest attention scores play an important role in predictions of self-attention models. Hence these are chosen as the possible vulnerable words. These words are greedily perturbed until the attack is successful. One of the methods "Textfooler" [52] uses a similar strategy where important words are greedily replaced with synonyms until the classification label changes. Another work in this direction [46] proposed TextExplanationFooler algorithm, which designed word importance-based attacks for explanation models in text classification problems. Working in a black box attack setting proposed attack attempted to alter outputs of widely used explanation methods while not changing the predictions of the classifier.
- In replacement-based methods, words are randomly replaced with semantically and syntactically similar words. Here the replacement for words is obtained by using word vectors like GloVe [99] or thought vectors. [64] used

thought vectors to map sentences to vectors and replaced one word from it's nearest neighbors which had best effect on the objective function. [3] used GloVe vectors to randomly replace words that fit in context of sentence.

Sentence level adversarial attacks These attacks can be considered as manipulation of a group of words together instead of individual words in the sentence. Moreover, these attacks are more flexible, as a perturbed sentence can be inserted anywhere in the input, as long as it is grammatically correct. These attack strategies are commonly used in tasks such as Natural Language Inferencing, Question-Answering, Neural Machine Translation, Reading Comprehension, text classification. For sentence-level attacks novel techniques such as ADDSENT, ADDANY are introduced in literature in recent years with variants such as ADDONESENT, ADDCOMMON [49, 141]. Some of the sentence based attacks are created such that they don't affect the original label of the input and used as a concatenation in the original text. In these cases, the correct behavior of the model is to retain the original output and the attack is successful if the model changes the output/label. In another set of methods, GAN based sentence level adversaries are created which are grammatically correct and semantically close to the input text [170]. Another example "AdvGen" [19] is introduced which is an example of gradient based white-box method and used in neural machine translation models. They used greedy search guided with the training loss to create the adversarial examples while retaining semantic meaning. Another work in this direction, [47] proposed syntactically controlled paraphrase networks (SCPNS) for adversarial example generation where they used encoder-decoder network to generate examples with a particular syntactic structure.

Multi-level adversarial attacks Multi-level attack schemes consist of a mixture of some of the methods discussed above. These attacks are used to make the inputs more imperceptible to humans and to have a higher success rate. Hence, computationally more intensive and more complicated techniques such as FGSM have been used to create adversarial examples. In one such method, they create hot training phrases and hot sample phrases. In this method, the training phrases focus on what and where to insert, modify or delete by finding hot sample phrases in white and black box settings where deviation score is used to find the importance of the words [83]. Another example used "HotFlip" [29] which is a character level white-box attack swapping characters based on the gradient computation. Similar to many other techniques, TextBugger [74] tries to find the most important word to perturb using a Jacobian matrix in a white box setting. The important words after identification are used for creating adversaries by inserting, deleting and swapping along with Reinforcement Learning methods with an encoder-decoder framework.

3 TAXONOMY OF ADVERSARIAL DEFENSES

In this section, we will discuss the different types of defense methods used to protect deep learning models from adversarial attacks. We will also highlight some of the recent research works that have shown promise in this area. Adversarial defense strategies are methods used to prevent deep neural networks from failing due to adversarial attacks. These methods aim to increase the robustness of neural networks by training them in an environment that simulates adversarial attacks, or by adding mechanisms to detect and handle adversarial inputs. Another approach to increase robustness is to create a perturbation-resistant region around the input space. Therefore, in NLP, there are three main strategies for designing adversarial defense methods: creating a similar environment during neural network training, identifying malicious inputs during training and correcting them using specialized methods, and certifying the robustness of the input region for the network.

The defense methods in NLP discussed in this paper are divided into three main categories: adversarial training based methods, perturbation control based methods, and certification based methods, along with some miscellaneous approaches. Methods that do not fall under the first three categories are included in the miscellaneous category. The

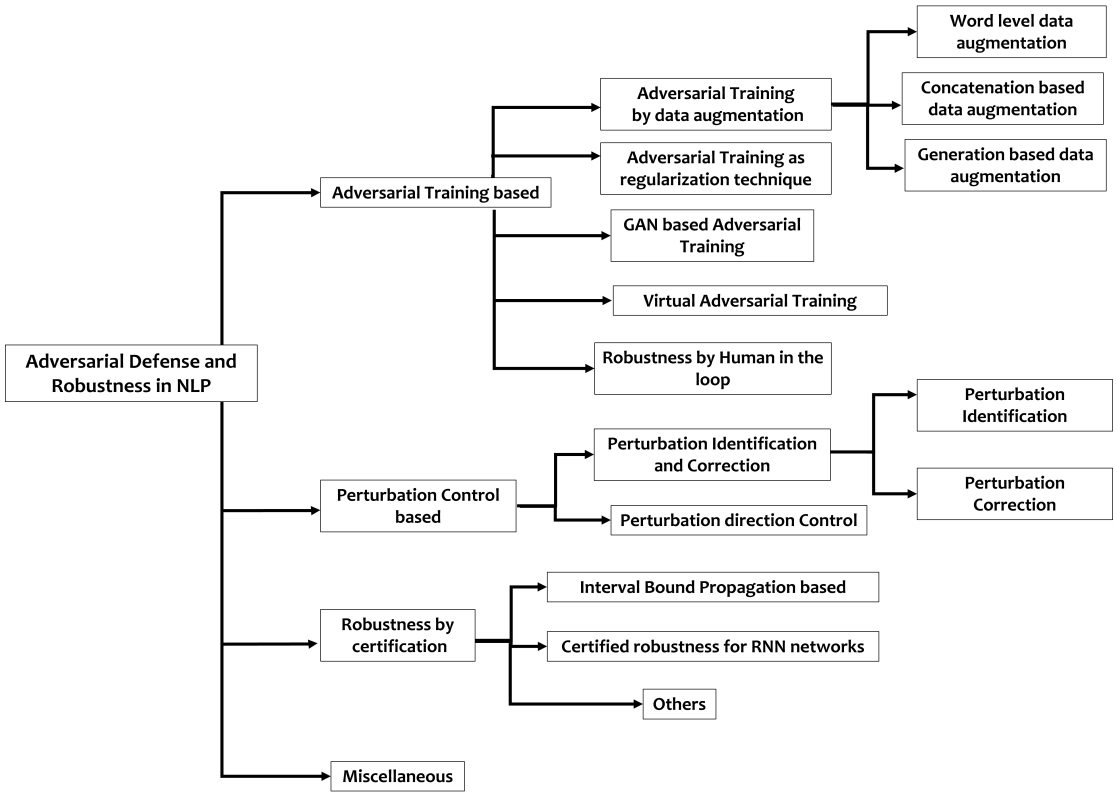

Fig. 1. Taxonomy of adversarial defense methods in natural language processing

first set of techniques falls under the category of (i) *Adversarial Training*, which serves as a defense mechanism against adversarial attacks. The various subcategories of methods included in this group are: Adversarial training through data augmentation, Adversarial training as a regularization technique, Adversarial training using Generative Adversarial Networks (GANs), Virtual adversarial training, and Robustness through human intervention. The second set of methods is based on (ii) *perturbation control*, and includes two subcategories: Perturbation identification and correction, and Perturbation direction control. The third set of methods follows approaches that provide (iii) *certification* of the model's robustness against adversarial attacks. The last set, (iv) *miscellaneous*, comprises a combination of various methods that do not fit into any of the aforementioned categories. Figure 1 illustrates the detailed taxonomy proposed in this survey article for adversarial defense techniques in NLP.

4 ADVERSARIAL TRAINING BASED DEFENSES

Adversarial training was first introduced in the work proposed in [35]. It is a method of defending against adversarial attacks by introducing adversarial examples in the training data. The strength of adversarial examples decides the final robustness and generalization achieved by the model. Adversarial training is further divided into sub-groups on the basis of the strategies used for augmenting the data such as word or character level modification, model-based generation of adversarial examples or adversarial inputs generated by concatenation in the original data. While some

of the methods performed adversarial training by generating a set of adversarial examples and inserting them in the training dataset, other methods used adversarial training as regularizer by introducing perturbation within the network. In this section, some of the work in literature will be highlighted which proposes to use of data augmentation for generating adversarial examples for adversarial training.

4.1 Adversarial training by data augmentation

Adversarial defense methods often use adversarial training as a basic technique for defending against adversarial attacks. This method involves generating a set of adversarial data using a perturbation scheme and incorporating it into the model training process. Many techniques for generating adversarial examples using data augmentation have been proposed in the literature. These techniques involve identifying the most important words, characters, or other parts of the input text that affect the output the most and manipulating the input data by flipping, inserting, deleting, or swapping those parts of the sentence. Concatenating a piece of text in the input by finding the most appropriate position is another strategy used in the literature . There are also automatic methods for generating adversarial examples for adversarial training, some of which will be discussed in the following sections.

4.1.1 Word level data augmentation. There are methods in literature that propose to modify words or word embeddings in the input text for generating adversarial examples to augment data. In this line the work proposed in Textbugger [74] presents adversarial training by data augmentation for text classification in both white box and black box settings by generating utility-preserving adversarial examples. In the white box, important words for perturbations are found using Jacobian of classifier and then optimal perturbations are found by searching the nearest neighbor space in Word2Vec embeddings. In black box setting, data is augmented by finding important words and sentences which contribute to the output the most and manipulating them. The work in [20] proposed novel data augmentation technique for adversarial training in machine translation task, which reinforces the model to virtual data points around the observed examples in training data. They proposed vicinity distribution for adversarial space (space of adversarial examples centered around each training example), and sampled virtual adversarial samples from it using interpolated embeddings of existing training samples. In the work [78] adversarial perturbations are applied on word embedding layer of a CNN for text classification task to make the classification model robust towards the worst perturbations. Another work [155] proposed a new method PQAT which perturbs the embedding matrix rather than the word vector for machine reading comprehension task. Two additional independent embedding spaces for paragraph-question(PQ) are used, to give additional context for the same word used with different roles in paragraphs and questions. During training P-Q embeddings are added to the original vector keeping the context from passage and question separate.

In another work, [166], the authors proposed continuous bag-of-word (CBOW) embedding based perturbations for generating human imperceptible adversaries for text classification task. Embedding space generated by CBOW is used for predicting the perturbation direction and tries to preserve the meaning of the sentence by placing a constraint on the perturbation direction. Authors generated adversarial examples without altering the semantic meaning of the sentence and used these examples in adversarial training. In the same line, [84] proposed a solution to the Out-Of-Vocabulary (OOV) words problem faced by conventional character level defense methods leading to a poor performance of models. They proposed adversarial stability training to overcome these challenges. Stability training is a technique which makes the output of the neural networks significantly robust, while maintaining the original performance [171]. Proposed Adversarial Stability Training (AST) is used with character level embeddings, to overcome OOV problems and adversarial sentences are generated by perturbing each word, using character level embeddings representation

to overcome the distribution problem. Another work in the same direction, [43] created adversarial examples using several schemes such as, random word replacement, synonym replacement, finding weak spots in the input strings with greedy approach, by constraining the embeddings within L_1 distance, replacing the word on the basis of attention score (high attention score word to low score word) and demonstrate their results on sentiment analysis, textual entailment, and machine translation tasks. They analyzed the robustness of RNNs, transformers and BERT-based models and demonstrated that self-attentive models are more robust than RNNs. Work proposed in [9] evaluated adversarial training with adversarial examples for eight datasets in NLP targeted for several purposes such as question answering, reasoning, detection, sentiment analysis, and language detection using language models such as LSTM and GRU. They used combinations of dropout and adversarial example for evaluation. Another work [123] proposed data augmentation for adversarial training for increasing the robustness of causal reasoning task. They proposed data augmentation by synonym substitution and by filtering out casually linked clauses in the larger dataset and used generative language models to generate distractor sentences as potential adversarial examples. To improve the conventional adversarial training methods [138] proposed to use gradient based approach for ranking the important words in the training dataset and distilBERT similarity score for finding similarity between two-word embeddings for a faster and low-resource requirement-based adversarial training. Also they propose to use a percentage of training data for generating adversarial examples instead of converting all the training data for cheaper adversarial training.

In [28], authors have proposed a white box based defense method by generating adversarial examples with flip, swap, insert and delete at character level and used gradient based optimization method to rank the examples. The work [165] proposed Metropolis-Hastings (MH) sampling [21] based adversarial example generator for text classification. Three level word operations, replacement, insertion and deletion are performed with MH sampled words in a black box setting, while the gradient of the loss function is inserted in the pre-selection function of MH in case of white box attack. For the text classification task, in the work [26] adversarial training is used for cross lingual text classification and robustness enhancement by training it on English data and using the model to predict on non-English unlabelled data. The predicted outputs are used as adversarial examples for adversarial training. In the same line, [109], authors have proposed a greedy algorithm probability weighted word saliency for adversary generation for the text classification task. Adversarial examples were generated with word synonym replacement and named entities with other named entities using WordNet, picking up the words which cause maximum change in text classification probability. In the work [160], black box adversary generation is proposed which uses sememe (minimum semantic unit in linguistics) based word substitution that is more sememes per sense mean fewer substitute words that share the same sememes can be found, which negatively affects the adversarial attack success rate. Later they used Particle swarm optimization [59] based algorithm to search the optimal adversarial example. In literature, frameworks and APIs have also been proposed which provides a complete platform to the user with various kinds of attacks to generate adversarial examples to be used for adversarial training for defense. In this line, the work [91] presented a python framework for attack generation and defending the model, which augments the data using a word embedding, word swap, thesaurus word swap, homoglyph character substitution, etc. They also provide a set of constraints on the generated perturbations to keep them indistinguishable from the original such as, grammar check, POS tags etc. Along the same lines, [130], generated imitation models for Google API of machine translation and generated adversarial samples using techniques such as flipping, replacing malicious nonsense, substituting phrases that cause incorrect translation and attacking original model in a black box manner. Later they used those adversarial examples to train the imitation model and transfer the examples to the victim model. In this work the authors aim towards, finding vulnerability in the victim model to make it more robust, by having the victim model output a different high-accuracy translation. Apart from

word level perturbations, there are some sentence level perturbations proposed to generate adversarial examples, which are discussed in coming section. These adversarial examples are hand crafted and generated by identifying vulnerability of model, and further used with adversarial training to defend model from such attacks. However, some of papers proposed model based generation of adversarial examples are also discussed in coming sections.

4.1.2 Concatenation based data augmentation. Another approach to generating adversarial examples for data augmentation involves using concatenation-based strategies and automatic generation of adversaries with language models. For instance, [50] developed concatenative adversarial perturbations, AddSent, to generate adversarial examples for reading comprehension systems. This method involves concatenating grammatically correct sentences to end of a paragraph, which look like questions or other arbitrary sentences. They also replaced some answers with fake answers that have the same part of speech type and category. Similarly, [142] proposed AddSentDiverse, which generates adversarial examples with significantly higher variance by varying placement of perturbations. They also expanded set of fake answers used in AddSent and demonstrated the limitations of the original AddSent method.

4.1.3 Generation based data augmentation. Another approach to improving the robustness of machine learning models is through generation-based adversarial examples. For example, [56] proposed using knowledge-guided rules and a seq2seq model to generate new hypotheses from a given premise for textual entailment models. In a similar vein, [39] proposed a defense strategy using adversarial training, using a seq2seq model to generate perturbations for structure prediction tasks that involved predicting POS tags, parse-trees, and noun phrases. In [150], the authors proposed a grey box adversarial attack for sentiment analysis that uses a generator model for data augmentation. Adversarial training was conducted by augmenting data using a static copy mask mechanism in the generator, and counter-fitted word embeddings and label smoothing methods were used to better capture lexical relations and preserve the labels of adversaries. [92] introduced the idea of injecting antonymy and synonymy constraints into vector space representations. In the same line, [134] proposed a new method of adversarial example generation by controlled adversarial text generation where they aimed to perturb input for a given task by changing other controllable attributes of the dataset. For example, in the case of sentiment analysis task for product reviews, product category becomes a controllable attribute that cannot change the sentiment of a review. Their pretraining module consists of an encoder-decoder architecture, which is used to teach the model to copy the input sentence S, assuming that it has the controllable attribute (a) in the sentence. The decoder module is updated to generate a sentence containing attribute $a' \neq a$. In the optimizing module the subspace of all a' is checked by computing the cross-entropy loss to find the highest perturbation. In similar lines of developing robust NLP model, the work proposed in [65] studied a group of linguistic rules to demonstrate local semantic robustness within a sentence and generated variations in input text using predefined template with fixed label. These templates adhere to the linguistic rules discussed in the paper, where incoming variations will not be able to change the output label and further used them for training robust sentiment analysis models. In the direction of paraphrasing-based adversarial instance generation, [48] proposed syntactically controlled paraphrase networks (SCPNs), which generate a paraphrase of the given sentence with the desired syntax in a controlled manner. SCPNs use a bidirectional LSTM and a two-layer LSTM augmented with soft attention over the encoded states based encoder-decoder architecture, utilizing a paraphrase pair and a target syntax tree as the inputs. Using the adversarial instances in adversarial training, they evaluated the proposed method on sentiment classification and textual entailment applications.

Table 1 shows the summary of all the adversarial training methods with data augmentation. It summarizes the strategies used in the literature for perturbation generation, along with the granularity of the perturbations used,

Strategy	Work	Granularity	Application	Threat Model
Word	[74]	Words & Sentences	Sentiment Analysis	White Box, Black Box
	[20]	Sentence level	Machine Translation	Black Box
	[78]	Words	Text Classification	White Box
	[155]	Embedding Matrix	Reading Comprehension	White Box
	[92]	Word embedding	Dialogue state tracking	White Box
	[166]	Word embeddings	Text Classification	White Box
	[84]	Character Embeddings	Text Classification	White Box
	[43]	Word	Sentiment Analysis, Textual Entailment, Machine Translation	White Box
	[9]	Word and Character embeddings	Question Answering, Reasoning, Sentiment Analysis, Language detection	White Box
	[123]	Word level	Causal Relation classification	White Box
	[28]	Character level	Machine translation	White Box
	[165]	Word level	Text classification	Black box
	[109]	Word level	Text Classification	White Box
	[160]	Word level	Text classification	Black Box
	[130]	Phrase level	Machine Translation	Black Box
	[26]	Word level	Document & Intent classification	White Box
Concatenation	[50]	Sentence	Reading comprehension	White Box
	[142]	Sentence	Reading comprehension	White Box
Generative	[56]	Sentence generation	Textual Entailment	White Box
	[39]	Sentence generation	Predicting POS tags, parse trees, NP	Black Box
	[150]	Sentence generation	Sentiment Analysis	Grey Box
	[134]	Sentence generation	Sentiment Analysis	White Box
	[65]	Sentence generation	Sentiment Analysis	White Box
	[48]	Sentence generation	Sentiment classification, Textual entailment	White Box

Table 1. Summary of the adversarial training methods by data augmentation

demonstrated NLP applications, and the kind of the threat model they are defending. Many of the word-based methods for data augmentation involve operations like flipping, swapping, inserting, deleting, and synonym substitution with words to modify the original inputs and generate adversarial examples for use with adversarial training. Some methods use these operations with characters in the sentences instead of whole words. These adversarial perturbation generation methods aim to maintain the naturalness of the input text so that the perturbation is imperceptible to humans. Therefore, some of these methods use word embeddings for perturbations instead of words from the input text. Another set of methods in the literature use concatenation operations within input sentences by identifying the appropriate position for concatenation, and these adversarial examples are used in adversarial training. In addition to manual perturbations and adversarial example generation, another direction of methods uses generative models to create adversarial examples or perturbations. Along with the data augmentation methods, adversarial training methods introduce perturbations within the training loss functions, which are discussed in the next section.

4.2 Adversarial training as regularization technique

In this section, another defense method based on adversarial training for NLP is discussed. In this style of adversarial training, the input perturbations are incorporated as a part of model training, instead of training it with adversarial examples. In the work [35], authors proposed adding perturbations in input as a regularizer in the loss function. The modified optimization function based on the fast gradient sign method after adding perturbations is defined as:

$$L_{adv}(x_l, \theta) = D[q(y|x_l), p(y|x_l + r_{adv}, \theta)]$$

$$r_{adv} = \underset{r;||r||_2 \leq \epsilon}{\text{argmax}}\, D[q(y|x_l), p(y|x_l + r, \theta)]$$

Where, L_{adv} is the adversarial loss term, r_{adv} is the adversarial perturbation, x_l is the labeled input data, D is the non-negative divergence measurement function between two probability distributions, ϵ is the upper bound on the perturbations, $q(y|x_l)$ is the unknown true distribution of the output label. This loss function is designed to approximate the true distribution $q(y|x_l)$ by a parametric model $p(y|x_l, \theta)$ which is robust against adversarial attack to the input data x. Adversarial training using perturbations in the loss function has been shown to be a successful defense strategy as it generates adversarial examples that are difficult to create manually by exploiting flaws in the optimization function to improve model generalization. In the field of NLP, several techniques have been proposed to introduce perturbations during training or modify the loss function. Here, we describe some of the methods that have been proposed to introduce perturbations during training.

In the work [89], the authors included perturbations at each step of training and tried to minimize the loss function for text classification tasks. In another work [156], authors proposed adversarial training for POS tagging, by using character level embeddings with BiLSTM models, where word-level embeddings are generated by concatenating character level embeddings. Perturbations are added to the input at the character level embeddings in the direction which maximises the classifier loss function while training the model. Another method [146], introduced adversarial noise at embedding (concatenation of word and characters) level for the task of relation extraction within the multi-instance multi-label learning framework. They proposed a joint model to entity recognition and relation extraction using adversarial training as a regularization scheme where worst case perturbations are added to maximize the training loss. Along the same line, [133] authors improved the neural language modeling by using adversarial training as a regularization technique. They injected an adversarial perturbation on the word embedding vectors in the softmax layer of the language models while training the model. Authors have suggested new loss functions and diverse neural network optimization methods to train a model adversarially and improve its robustness. In this direction, the work [88] proposed adversarial training for natural language inferencing, by reducing the adversarial example generation problem to combinatorial optimization problem. They proposed a continuous inconsistency loss function that measures the degree to which a set of examples can cause a model to fail. By maximizing the inconsistency loss and constraining the perplexity of the generated sentences, adversarial examples are generated, posing it as an optimization problem. In the same direction, [57] proposed defense mechanism, diversity training, for transfer based attacks for the ensemble of models. They proposed gradient alignment loss (GAL) which is used as regularizer to train an ensemble of diverse models with misaligned loss gradients. Another work [25] proposed a novel Adversarial Sparse Convex Combination (ASSC) method to leverage regularization term for introducing perturbations. They modeled the word substitution attack space as a convex hull of word vectors and further proposed ASSC-defense for using these perturbations in adversarial training.

There are methods in the literature that proposed novel regularization techniques for enhancing the robustness of a language model. In this line of work [132] proposed InfoBERT for increasing the robustness of BERT based models by

analyzing language models from information-theoretic perspective. They presented two mutual information based adversarial regularizers for adversarial training. Information Bottleneck regularizer which extracts minimal features for downstream tasks and removes noisy and vulnerable information for potential adversarial attacks. Anchored Feature regularizer extracts strong local features which are not vulnerable while aligning local features with global features to increase the robustness. Another work [175] proposed FreeLB, to use K-step PGD [4] for generating adversaries in adversarial training and used multiple PGD iterations. In contrast with k-step PGD and freeAT [120] methods, used multiple PGD iterations to create adversaries simultaneously accumulates the "free" parameter gradient. In this method while optimizing the objective function, it replaces the batch of input with K times large batch which include perturbations along with inputs. Improving the performance of this work [81] proposed FreeLB++, by extending the search region to a larger l_2-norm and increasing the number of search steps at the same time since FreeLB has a narrow search space. They also bench-marked various defense methods in literature under standard constraints and settings to have a fair comparison of these methods. Table 2 describes the summary of adversarial training as regularization technique. In the coming section GAN based adversarial defense methods are discussed, where GAN is used as a generator and discriminator model to build more robust models against adversarial attacks.

Work	NLP task	Granularity
[89]	Text Classification	Word embedding
[156]	POS Tagging	Character embeddings
[146]	Relation Extraction	Word and character embeddings
[133]	Machine translation, language modeling	Word embeddings
[88]	Natural Language Inferencing	Sentence Embedding
[25]	Sentiment Analysis and Natural Language Inferencing	Word embeddings
[132]	Question Answering, Natural Language Inferencing	BERT embeddings
[175]	Natural Language Inferencing, Textual Entailment	Word embeddings
[81]	Sentiment analysis, Text Classification	Word embeddings

Table 2. Summary of adversarial training as regularization technique

4.3 GAN based adversarial training

Using GAN for adversarial training is another approach that has been used in literature for defending models from adversarial attacks. In this approach a generator and discriminator are trained together in adversarial manner, where the generator is primarily used for generating adversarial examples. The discriminator is responsible for discriminating between the clean data and the adversarial training samples to make model more robust towards adversarial attacks. In this line, the work proposed in [56] uses adversarial training withva GANs for the textual entailment task. The generator and discriminator are trained in an end-to-end manner, where generator (seq2seq) is trained for generating adversarial examples using external knowledge or handwritten rules. Discriminator is trained in the same manner to learn the textual entailment for the generated samples. In another work, [110], authors used a conditional variational autoencoder which generates the adversarial examples for text classification task. They further used a discriminator and GAN training framework for adversarial training and to make sure the generated adversaries are consistent with the real-world data. In the work [85] authors used multi-task learning for domain adaptation. They proposed a model which has separate units to discriminate between shared patterns and task specific patterns using GAN for creating adversarial examples and includes this loss in the final loss for optimization. They demonstrated the improved performance over 16

datasets and the learned parameters in both shared and task-specific parts of the network. The work in [149] proposed a new adversarial training approach that mimics a GAN. The generator is used to create adversarial examples with the help of lexical knowledge base where a classifier is used to score the generated adversarial example. The model is trained in a reinforcement learning fashion due to discrete-time generation of the generator model and the score from the classifier is used as the reward for the generator. The generator generates examples by replacing words in the input sentence with synonyms, neighboring words, and superior words. In another work, [22] try to defend against an attacker who tries to take the encoded information to reconstruct the original input text. In the adversarial training-based defense strategy they used GAN style training with 2 components in which the original one predicts the class of a given sentence and a binary classifier predicts the privacy element. The training style involves making the output prediction as correct as possible while at the same time creating more complex examples for the privacy element classifier. Table 3 describes the summary of GAN based adversarial training methods, where each row describes the NLP task for which defense is designed and the specific strategy used in the proposed work.

Work	NLP Task	Strategy
[56]	Textual Entailment	Using external knowledge
[110]	Text classification	Using Conditional Variational Autoencoder (VAE)
[85]	Text classification	Adversarial shared-private model
[149]	Sentiment Analysis	Using lexical knowledge base
[22]	Sentiment Analysis and Topic classification	Privacy measurement of neural representations

Table 3. Summary of GAN based adversarial training methods

4.4 Virtual Adversarial Training (VAT)

Virtual Adversarial Training (VAT) is another variant of adversarial training-based defense methods first proposed by [90]. VAT is found to be a very efficient method in the case of semi-supervised learning methods because it defines the adversarial direction without label information. In contrast to Adversarial training, VAT doesn't need full label information for generating perturbation. The intuition behind VAT is to add perturbation $'r'$ to input x such that the divergence of their output space is maximum. Hence, training is done in a way to minimize the divergence term after adding the perturbed input to make the model robust against adversarial attacks. The modified loss function for virtual adversarial training is defined as:

$$LDS(x_*, \theta) = D[p(y|x_*, \hat{\theta}), p(y|x_* + r_{adv}, \theta)]$$

$$r_{adv} = \underset{r;||r||_2 \leq \epsilon}{\operatorname{argmax}} D[p(y|x_*, \hat{\theta}), p(y|x_* + r_{adv})]$$

$$R_{adv}(D_l, D_{ul}, \theta) = \frac{1}{N_l + N_{ul}} \sum_{x_* \in D_l, D_{ul}} LDS(x_*, \theta)$$

Where, x_* are the "virtual" labels which are probabilistically generated and r_{adv} are virtual adversarial perturbations. Here, x_* is kept in the place of x, since label information is not available for all the input data. LDS is the local smoothness term for the input data point x and R_{adv} is the final regularization term.

The work proposed in [89] extended the notion of virtual adversarial training and adversarial training for text classification and sequence models proposing this technique as a regularization method. For defending the models, they introduced perturbations in word embeddings of the text inputs, while minimizing the KL divergence of VAT. In another

work [100], authors proposed a VAT method by performing adversarial steps on those examples which are predicted as wrong by the model and then regularise the model for this target direction in contrast with general adversarial training methods where the perturbation is done for all examples with variation from the gold label. In a targeted training manner, they try to steer the examples to a particular label y_t and presented a comparison with human-annotated data along with other adversarial training algorithms. In the same direction, the work [86] authors proposed a novel adversarial robust model "Adversarial training for large neural LangUage Models(ALUM)" for defending BERT-based pretraining language models. It is a general model for adversarial training in pretraining and fine-tuning which regularizes the training objective by applying perturbations in the embedding space that maximizes the adversarial loss. The model is regularized using VAT. Experimenting with different word embeddings using VAT, [166] extended the adversarial training regularization for semi-supervised tasks. They used continuous bag of words (CBOW) model for generating word embeddings and restricted perturbation directions for creating adversaries. Targeting specifically sequence labelling tasks in NLP, [18] proposed VAT for sequence labelling task combining CRF, making sequence labelling task more robust. They use CNN layer for extracting character and word embeddings, LSTM for sequence encoding, and CRF decoder layer to incorporate the probabilities of label transition. Introducing more variations to VAT, [77] proposed Token aware virtual adversarial training. In contrast with conventional virtual adversarial training, TAVAT generated token aware perturbations instead of random perturbations to avoid unnecessary noise and take important information carried by tokens into consideration. Table 4 describes the summary of Virtual Adversarial Training (VAT) based methods along with the specified NLP task for their design and granularity of the perturbation.

Work	NLP task	Granularity
[89]	Text classification & Sequence modeling	Word embeddings
[100]	Natural Language Inferencing (NLI)	Word embeddings
[86]	Question answering, NLI, Named Entity Recognition(NER)	BERT embeddings
[166]	Sentiment classification	Word embeddings
[18]	Chunking, NER, slot filling	Character and word embeddings
[77]	NER, NLI, Textual Entailment, text classification	Token level

Table 4. Summary of Virtual Adversarial Training (VAT) based defense methods

Along with the several variants of adversarial training methods, there are a few schemes that utilizes Human-In-The-Loop (HITL) framework, where human-level intervention is considered for adversarial training. Some of these methods are discussed in the next section.

4.5 Robustness by human in the loop

Human-In-The-Loop (HITL) is an idea of leveraging human intervention while training models or defending them against adversarial attacks. The scheme is extensively used in various fields of artificial intelligence. It takes advantage of both human and machine intelligence, for labeling the data, and training and validation of models. While it is proven to be an efficient scheme in other areas of artificial intelligence, several authors have tried to use HITL for developing algorithms for adversarial defenses.

The work [160] proposed a sememe-based word substitution method to generate perturbations. Apart from using particle swarm based optimization algorithm to search perturbation for data augmentation, they manually selected 692 valid adversarial samples for adversarial training to further boost the performance. Also, authors in [94, 144], created dataset of adversarial examples, ANLI for natural language inferencing task by crowd-sourcing. Adversarial examples

are written and verified by human annotators in 3 stages in loop, while getting them tested from high performing NLI models. They also presented error analysis and discuss annotation scheme and data collection process of ANLI. In the work [24] authors built a model for offensive language detection in dialogues using human and models in loop. They trained BERT based model on Wikipedia Toxic Comments dataset, and asked crowd workers for marking the messages as offensive if they are wrongly marked safe by the system. This process is performed in multiple iterations to build a robust system. Work in [129] presented a defense method using Human-In-The-Loop by proposing human-computer hybrid approach for evaluating the models. They presented a human verification of the question-answering system, where human annotators authored adversarial examples to break a model-based QA system but still answerable by humans. This process is targeted towards building a robust question-answering system by inserting human-authored adversarial examples. Table 5 depicts summary of the Human-In-The-Loop (HITL) based adversarial training methods in NLP defense. Each row in table depicts the NLP application for which the defense method is designed and granularity at which human interaction is proposed. Adversarial training-based defense methods are evidently the most popular way to build robust models against adversarial attacks in NLP. It requires the generation of adversarial examples using several techniques and using them for training the model. However, in contrast with training with adversarial examples, there are methods in the literature that are proposed for the detection of perturbations, their correction, and controlling directions of those perturbed instances. Some of these methods are discussed in the next section.

Work	NLP Tasks	Human interaction granularity
[160]	Sentiment Classification, NLI	Manual selection of word substitution based perturbations
[94]	NLI	Crowd Sourced and tested with models
[144]	NLI	Crowd Sourced and tested with models
[24]	Offensive language detection	BERT with crowd workers
[129]	Question Answering	Human annotators and verification

Table 5. Summary of the Human In The Loop (HITL) based adversarial training methods

5 PERTURBATION CONTROL BASED DEFENSES

The adversarial defense methods proposed in previous sections, use data augmentation schemes, perturbation generation within training for supervised and semi-supervised tasks, adversaries monitored by human and models in loop, defending the model in a generating and discriminating manner. However, all these schemes do not incorporate the idea of interpretable perturbations or reconstruction of generated perturbations. In literature, there are schemes that control direction of perturbations to make the perturbations more meaningful, indistinguishable and re-constructive and further use them in training. Also, another set of method try to identify the perturbed inputs and correct them to make the models more robust. In this line, the following sections describe methods which have been proposed in this direction.

5.1 Perturbation identification and correction

In literature perturbation, control-based method tries to identify malicious inputs after an attack and correct them. Some of the methods only identify these inputs and filter the training data accordingly. While the other category of methods also corrects these inputs after their identification using methods such as spell checker or rule-based. Some of these methods are described below.

5.1.1 Perturbation Identification. Perturbations related to word modifications which included insertion, deletion, substitution or swapping of words are identified in several ways. One of those methods is proposed in [139], where the

authors proposed defense mechanism, against synonym substitution, calling it “Synonym Encoding Method”(SEM). They essentially clustered all the synonyms in embedding space with their euclidean distances and then encoder is layered before input to train the model. Encoder is responsible for identifying all the synonym substitution-based attacks in the model and maps all the synonyms to a unique encoding without adding extra data for training. Improving on this work [154] proposed a robust adversarial training method called Fast Triplet Metric Learning (FTML). This method tries to cluster the similar embedding and pushes dissimilar embedding where SEM works directly with input texts and establishes no relation with non synonym clusters. FTML forces each word with a similar meaning to have the same representation in the feature space and pushes away the word with a different meaning. The method incorporates a word level triplet loss which tries to minimize the distance between a word with its corresponding group of synonyms and maximizes the distance with its non-synonym group. In another work in this direction, [173] proposed Dirichlet Neighborhood Ensemble (DNE), a randomized smoothing method for training a robust model to defend substitution-based attacks. DNE forms virtual sentences by sampling embedding vectors for each word in an input sentence from a convex hull spanned by the word and its synonyms, and it augments them with the training data, (mixing the embedding of the original word in the input sentence with its synonyms). The work in the same line [6] introduced frequency-aware randomization framework Anomaly Detection with Frequency Aware Randomization (ADFAR) for defense against adversarial word substitution. They add an extra module to detect perturbation in the sentences and apply ADFAR only on sentences that are identified as adversarial. This module is added to the language model and use a multi-task learning procedure. They demonstrated that ADFAR works better than other defenses on 4 datasets - MR, SST-2, IMDb, MNLI. The work [140] proposed an adversarial defense scheme by perturbation detection for synonym substitution attacks. They proposed a novel method, namely, Randomized Substitution and Vote (RS&V). The proposed method calls an input text “adversarial example", by randomly substituting some of its words by their synonyms and checking the consistency of the highest voted label for all perturbed examples. If the label is found to be inconsistent with the original label then it is considered as an adversarial input. Working in a similar direction, [127] proposed to use of random perturbations to defend sentiment analysis models. They inserted random perturbations to the multiple copies of a randomly selected sentence from the reviews. These perturbations included synonym substitution, dropping of a word or a spell check with correction if necessary. All these perturbed copies of the sentences are put together with the original review and a majority vote is taken for each sentence by the sentiment classifier. By majority-based classification, the models are taken back to their original performance accuracy before the attack. In the direction of perturbation identification, [147] proposed an extensive dataset, TCAB, for attack detection and labeling with over 1.5 million instances of adversaries. To construct this dataset, authors used 6 datasets for text classification with 3 target classifiers and 12 different attacks from textAttack and openAttack and presented a benchmark for these attacks. They proposed an attack identifier/labeler, using three features of the input such as text properties including contextual embeddings, length of the input, token case, punctuation, non ASCII characters. Authors use language model’s properties as another feature that identifies the structure of the language such as phrasing or ungrammatical input text. They used target model’s properties as another feature that captured the changes in model’s gradients, activation or saliency because of malicious inputs and trained a classifier with them. Utilizing the TCAB dataset and the text, language, and classifier features, the detection of the perturbation and labeling the type of attack is proposed. Another work [136] proposed a novel method called TextFirewall identification of adversarial inputs. They used word importance to quantify the importance of a word in the input sentence to the final classification of the model. The impact of each word in an input sentence is calculated to detect the perturbation, by summing the scores of each model, and then the model’s output is compared with original ground truth to identify the perturbed input. In

a similar direction, [158] proposed an adversarial robustness method by proposing feature density estimation-based perturbation detection. In contrast with the available frequency-based likelihood estimation, they utilized the probability density of sentences. They proposed a method of Robust density estimation (RDE), which fits the probability density estimation model on the features obtained from a pre-trained model like BERT. Dimensionality reduction is applied to the parameters of these features to avoid the curse of dimensionality. In addition, they released a benchmark for word-level adversarial detection using 4 NLP models with four different datasets for text classification. In addition to the discussed work in perturbation identification, some of the methods in literature also attempt to correct these perturbed input data after their detection. Some of the perturbation correction methods are discussed in next section.

5.1.2 Perturbation correction. Adversarial inputs are also required to be corrected after their identification to retain the training data. Some of the methods in this direction of work attempted to identify perturbations related to character-level modifications in the input text. In this direction, the work [116] authors proposed a semi-character level recurrent neural network (ScRNN), which act as a spell checker by recognizing words. ScRNN has an architecture similar to standard RNN and takes semi-character vector as input and predicts a correct word at each time step by applying three types of noises: jumble, delete, and insert. As an extension of the above work, in [102] authors try to combat misspellings by using a word classifier before the actual classifier of a task. They propose ScRNN with backoff, to overcome limitations of ScRNN [116], and propose three backoff techniques if the word classifier predicts it as unknown (UNK). As a backing-off step, the word recognizer either passes the UNK word as is, backs off to a neutral word or backs off to a more general word recognition model trained on a larger, less specific corpus. In the work, [58] authors demonstrated the limitations of spell checker for perturbation identification & correction. They proposed a method in which context independent probability distribution are created by segmenting the perturbed sentence using BERT tokens and modified version of levenshtein distance. For context dependent probability - all the embeddings of context-independent hypothesis are clubbed into a weighted embedding. Now a token is masked and MLM is used to predict the tokens. This process is repeatedly done for the best approximation. Now, these hypotheses are sent to GPT for getting the language modeling score and the best hypothesis is selected from that. They compared their method again Pyspellchecker, human annotations and RNN trained for spell checking. In the work, [31], authors presented backdoor attacks as adversarial attacks during training of the model and proposed attacking methods for NLG model by inserting trigger words in the input sentence. They further proposed defense strategies by detecting of hacked inputs and output correct results and preserving the correct input and giving its output.

In [172] proposed a novel method for perturbation identification and correction, in which they try to recover the perturbed token based on the context and with the help of small world graphs. First they use, BERT model to get the contextualised embedding vector for each token and then pass it to a binary classifier for classification of perturbation. Later they used a BERT-based context network, to be used as the context for predicting the perturbed word. The perturbed word is masked and passed to the BERT to get the embedding of the mask token. Using the embedding vectors and small world graphs they recovered the affected tokens. In the direction of perturbation identification and correction, In [10] various types of perturbed text are identified and corrected using rule based methods such as alternating characters defense which corrects the combined unicode, space separation, and zero-width space separation perturbation in the entire document. Another rule-based defense used is Unicode Canonicalization which corrects & replaces unicode and tandem character obfuscation perturbations. They further used a continuous bag of words based embeddings and identified embeddings which are generated by parts of a single word combining random spaced words followed by a process to find similar embedding for words with similar spelling. A skip-gram model is trained with

vectors of similar context to ensure the embeddings having similar spelling and context are closer. They evaluated their embeddings with the downstream task of classifying the Facebook posts as engagement bait or otherwise. In a different line, the work [174] proposed a universal perturbation detection method, TREATED to defend against various perturbation levels without making any prior assumptions. They utilized several reference models to make different predictions about clean and adversarial examples and block them if found adversarial. They designed the reference models on the basis of their consistency on the clean and adversarial data. In the direction of identifying perturbations for other language text than English, the work [72] proposed a defense model for text classification for Chinese language. Adversarial perturbations are detected in 3 steps. Neural Machine Translation (NMT) model is used for removing the noise in the input text. The corrected text is converted into multimodal embeddings (semantics, glyph, and phonetics) and the extracted features are given into text classification.

Table 6 demonstrates the summary of the various perturbation identification and correction methods. It shows the type of attack for perturbing the input data, the strategy used for detecting the perturbations, NLP applications on which the proposed method is demonstrated, and whether or not they are attempting to correct the perturbed input after their detection. As it can be seen that a large part of perturbation detection and correction method is limited to synonym substitution and misspelling-based adversarial attacks. Also, the demonstration of the proposed defense is largely demonstrated on various types of text classification tasks including sentiment classification, news category classification, and topic classification. The commonly used techniques for perturbation correction after their detection includes blocking the perturbed data, generating or predicting the clean text, and replacing it with similar correct words. There are methods proposed in literature which defend the machine learning model by controlling the direction of perturbations in their embedding space. Some of these methods are discussed in the next section.

5.2 Perturbation direction control

The proposed work under perturbation direction control alters the direction of the perturbations towards the cleaner text input limiting the adversarial space. Along this line, the work [119] proposed an interpretable adversarial training method by restricting the direction of adversarial samples. The direction of perturbation is restricted to the words in the existing vocabulary so that perturbations could be interpreted even after adversarial training. In the work [166] authors propose to use CBOW to predict the perturbation direction while trying to preserve the meaning of the sentence by placing a constraint on the perturbation direction. Another work, [111] proposed an adversarial defense mechanism, Sequence Squeezing, aimed to make RNN models and their variants robust against adversarial attacks. The proposed method generates semantic preserving embeddings which are low in the number of features than the original embedding. The squeezed embedding is tested for adversarial attacks in malware detection and added to the training data while diminishing the adversarial space for generating perturbations. Table 7 presents the summary of perturbation direction control based adversarial defense methods in NLP, where each row shows the NLP applications and the type of perturbation used in the threat model in the associate work. Proposing adversarial defense for text input data with perturbation direction control is a step towards developing more interpretable defense model than conventional methods of adversarial training. The discussed methods in this category demonstrate their defense scheme on various type of text classification tasks, such as sentiment classification, malware classification. Another direction of adversarial defense methods in NLP propose to provide a certified region of robustness while training their machine learning model. Some of these methods are discussed in detail in the coming section.

Work	Attack	Method	Application	Perturbation Identification	Perturbation Correction
[139]	Synonym Substitution	Maps synonyms to unique encoding	Sentiment, Topic classification	✓	–
[154]	Synonym Substitution	Cluster similar embeddings	Sentiment, Topic classification	✓	–
[173]	Substitution based	Mixing embeddings of words & synonyms	Sentiment, , News category classification	✓	–
[6]	Substitution based	frequency aware randomization	Sentiment classification, Natural Language Inference	✓	–
[140]	Synonym substitution	Randomized synonym substitution & vote	Sentiment, Topic, News category classification	✓	–
[127]	Synonym substitution	Random perturbation defense	Sentiment analysis	✓	–
[147]	Attacks from TextAttack [91, 161]	TCAB attack identification dataset & Text, Language, Classifier properties	Sentiment, Abuse/No-abuse classification	✓	–
[136]	Word level perturbations	Finding word importance	sentiment classification	✓	–
[158]	Word level perturbations	Feature density estimation	Sentiment, News categories, Topic classification,	✓	–
[116]	Misspellings	ScRNN- Spell checker	–	✓	✓
[102]	Misspellings	ScRNN with Backoff	Sentiment Analysis	✓	✓
[58]	Misspellings, orthographic attacks	Context independent probability distribution	Restoring sentences	✓	✓
[31]	Backdoor attacks	Trigger word manipulation and BERTScore	Machine translation, Dialogue Generation	✓	✓
[172]	Word perturbations	Recover perturbed tokens with small world graphs	Text classification	✓	✓
[10]	Misspellings	Embeddings similar to original words, Rule based methods	Engagement Bait Classifier	✓	✓
[174]	Synonym substitution, replacement order strategy	Using reference models	Sentiment analysis	✓	✓
[72]	Word level perturbations	NMT model is used for removing noise	Text classification	✓	✓

Table 6. Summary of defense schemes proposed for perturbation identification and correction

Work	NLP Taks	Perturbation
[119]	Sentiment classification	Word embeddings
[166]	Sentiment classification	Word embeddings
[111]	Malware detection	Word embeddings for API call command

Table 7. Summary of perturbation direction control based adversarial defense methods in NLP

6 ROBUSTNESS BY CERTIFICATION

The methods discussed in the previous sections for adversarial defenses involved word/character substitution-based adversaries where words are synonyms to make the perturbation look indistinguishable. Other methods tweak words by inserting characters, changing spellings, and deleting/swapping characters. All these adversarial samples are necessary for defending the models but they are not sufficient. An attacker can generate millions of adversarial examples by modifying every word in a sentence. A defense algorithm based on adversarial training requires a sufficient amount of adversarial data to increase the robustness, which still does not cover a lot of unseen cases which are generated by exponential combinations of different words in a text input. Also, perturbation control-based methods require identification of perturbations on the basis of already-seen perturbations with a prior assumption of the type of attack. These methods have limitations in their performance when model is exposed to new adversarial instances. Hence, there is a separate set of adversarial defense methods in the literature which are driven by “certification". These methods train the model to provide an upper bound on the worst-case loss of perturbations and hence provide a certificate of robustness without exploring the adversarial space.

6.1 Interval Bound Propagation based methods

Interval Bound Propagation (IBP) [37] is a bounding technique, extensively used in images for training large, robust, and verifiable neural networks. Training the neural networks with IBP technique tries to minimize the upper bound on the maximum difference between the classification boundary and input perturbation region. IBP lets you include the loss term in the training, using which the last layer of the perturbation region can be minimized and kept on one side of the classification boundary. Now, this adversarial region is tighter enough and can be said certified robust.

In this line, the work [51] proposed certified robust models while providing maximum perturbations in text classification. Authors used interval bound propagation to optimize the upper bound over perturbations. IBP gives an upper bound over the discrete set of perturbations over word vector space. IBP computes an upper bound on the model’s loss when given an adversarially perturbed input. This bound is computed in a modular fashion. In another work [44] introduced a verification and verifiable training of neural networks in NLP. Authors proposed a tighter over-approximation in the form of a ‘simplex’ in embedding space in the input to generate perturbations. To make the network verifiable they define it as the convex hull of the all the original unperturbed inputs as a space of delta perturbation. Using IBP algorithm they generated robustness bounds (by generating bounds for each layer). In the work, [157] proposed structure-free certified robust models which can be applied to any arbitrary model. This method overcomes the limitations of IBP based method in which they are not applicable to character level and sub-word level models. They prepared a perturbation set of words using synonym sets, top-K nearest neighbors under the cosine similarity of GLOVE vectors, where K is a hyperparameter that controls the size of the perturbation set. They further generated sentence perturbations using word perturbations and trained a classifier with robust certification. In the context of IBP methods, [131] demonstrates the lack of generalizability of IBP-based methods for novel contextual

embeddings and a wider range of NLP tasks. They demonstrated the performance of their method in the sentiment analysis task.

6.2 Certified robustness for RNN networks

Despite having a plethora of work in finding a certificate for robustness, there is a lack of applicability in RNN based network due to their inherent complexity. Hence, in another line of work for robustness by certification, certified robustness for RNN based networks and self-attentive networks is proposed. In this line, the work Popqorn [61] proposed certified robustness for RNN based networks such as LSTM, GRUs. The challenge is to find a certificate of robustness in RNN based networks with their complex feedback architectures, the sequential inputs, and the cross-nonlinearity of the hidden states. Authors used 2D planes to bound the cross nonlinearity in LSTMs and proposed to find a certificate within a l_p ball (attack distance) if the lower bound on the true label output unit is larger than upper bounds of all other output units. They generated certificate of robustness by writing all the bounds as a function of epsilon and tried to find the optimum value of epsilon using a binary search procedure. In the work Cert-RNN [27], they overcame the limitations of Popqorn [61] by a robust certification framework for RNNs. The method overcame the limitations of Popqorn by keeping the inter-variable correlation and speeding up the non-linearities of RNN for practical uses. Authors created a zonotope [30] around the input perturbations and used that to be passed through a vanilla RNN or LSTM. The properties of the output zonotope can be verified to be certifiably robust. They used zonotope instead of a box to preserve inter-variable correlation, the precision of the network, and achieve a tighter bound. They could achieve tighter bounds and at least 19 times faster framework than Popqorn. In this line, [168] proposed a novel approach, Abstractive Recursive Certification (ARC) for certified robustness in RNN based networks. Authors defined a set of programmatically perturbed string transformations and constructed a perturbation space using those transformations proposed in [165]. They memorized the hidden states of the strings in the perturbation space that shared a common prefix to reduce the evaluation of LSTM cells while finding an upper bound to the loss and avoiding re-computing of hidden states. Following that they represent all the perturbation sets as a hyperrectangle and pass the hyperrectangle through the remaining network using IBP technique [37]. Following a similar direction, the work in [114] presents Polyhedral Robustness Verifier of RNNs (PROVER) which represents the perturbations in input data in the form of polyhedral which is passed through a LSTM network to obtain a certifiable verified network for a more general sequential data. Another work in this direction is proposed by [12] where authors proposed DeepT an abstract transformer-based network certification method. They attempted to certify larger transformers against synonym replacement-based attacks. In this work, authors propose to use multi-norm zenotopes improving the precision of standard zenotope based methods which works well for longer sentences by certifying larger radii of robustness ($\times$28 of existing methods). In another work [121] proposed an algorithm for verifying the robustness of transformers with self-attention layers which include challenges such as cross-linearity and cross-positional dependency. They provide a lower bound to a boundary (delta certificate) which will be always greater than 0 (probability of correct class is always higher than incorrect class) within a set of inputs which also include perturbations and tighter than IBP. They achieved this by computing lower/upper bound for each neuron with respect to the input space.

6.3 Other convex optimization based methods

There are other methods in literature for finding certified robustness for neural networks which used several convex optimization schemes and randomized smoothing-based schemes. In this line, [124] certified defence method is proposed for text classification task. They consider data sanitation defences, which examine the entire datasets and try to remove

poisoning points. They upper bound the worst possible test loss of any attack which works in an attacker-defender setting at the same time. They generated a certificate of robustness (upper bound) by inserting perturbed data at the time of training where defender is learning to remove outliers at each iteration. Upper bound fits all possible points that evade outlier removal. In the work [104] authors proposed certified robustness method based on semi-definite relaxation. They computed an upper bound on the worst case loss of the neural networks with one hidden layer. The computed certificate of robustness provides an upper bound on the robustness for all kinds of attacks and being differentiable they trained it jointly with the network. The work [135] provided a certificate of robustness with the idea of differential privacy in the input data. They implemented differential privacy in the textual data by treating a sentence as a database and words as an individual records. If a predictive model satisfies a certain threshold (epsilon-DP) for a perturbed input, its input should be the same as the clean data. Hence providing a certification of robustness against L-adversary word substitution attacks.

Certification method	**Work**	**NLP Taks**	**Models**	**Perturbations**
IBP	[51]	NLI, Sentiment classification	Feed Forward network, CNN, Bi-Directional LSTM, Decomposable attention	Word substitution
	[44]	Sentiment & Topic Classification	1 layer convolution network	word substitution and character typos
	[131]	Sentiment classification	CNN	word embeddings
	[157]	Sentiment & Text classification	BERT	Word Substitution
RNN based	[61]	Question Classification	LSTM, GRU	ϵ bounded L_p ball
	[27]	sentiment analysis, toxic comment detection, and malicious URL detection	RNN, LSTM	ϵ bounded L_p ball
	[168]	Sentiment Classification	LSTM	Word substitution
	[114]	Speech classification	RNN, LSTM	ϵ perturbation, dB perturbation
	[12]	Sentiment Classification	Transformer networks	L_p noise where $p \in \{1, 2, \infty\}$
	[121]	Sentiment classification	Transformer	ϵ bound perturbations
Other methods	[124]	Sentiment Classification	SVM	ϵn poison points
	[135]	Text classification	LSTM	Word substitution
	[162]	Sentiment & Topic classification	BERT & RoBERTa	Word substitution & Character level
	[66]	Sentiment, News, Topic classification	CNN, LSTM	Word substitution
	[101]	Toxicity, Occupation classification	CNN, BERT	Word substitution

Table 8. Summary of the certifiable robustness methods in NLP

Another work, [162] proposed defense algorithm to overcome the limitations of previous methods with an assumption that perturbation generation methods will be known a priory. They proposed RanMASK, a certifiably robust defense method against text adversarial attacks based on a new randomized smoothing technique for NLP models. Manually perturbed input text is given to the mask language model. Random masks are generated in the input text in order to

generate a large set of masked copies of the text. A base classifier is then used to classify each of these masked texts, and the final robust classification is made by "majority vote" and trained with BERT and RoBERTa to generate and train with masked inputs. Another work in this direction [66] estimated the Maximum Safe Radius (MSR) for a given input text, i.e. minimum distance between the classification boundary and embedding space. They quantified the robustness of neural networks against word replacement which is based on a minimum safe radius. They approximated the upper bound using Monte Carlo tree search and the lower bound by constraint relaxation technique of MSR for CNN and LSTM networks. In [101], the authors also tried to club the concept of fairness and robustness to increase the robustness of a neural network. They demonstrated that a certified robust model can also be used as a bias mitigation system to build trustworthy NLP systems. They integrated a bias mitigation system with state-of-the-art certified robust models to improve the robustness of a model. Table 8 presents the summary of the robustness by certification methods in NLP where rows are grouped by the certification method used, and each row describes the NLP application, machine learning model used for certification and type of perturbations generated in the threat model with each associated work. Currently these methods are not proven to be generalized across different types of deep neural networks and they have been evaluated for small set of NLP tasks and on smaller networks. There are various methods for defending the neural network from adversarial attacks and achieving robustness which are not discussed in these sections and follows a different line of approach, described in the next section.

7 MISCELLANEOUS

In the previous section, various methods are discussed for adversarial defenses and robustness enhancement. These methods follow the proposed taxonomy and categorization of adversarial defence schemes discussed in Sec. 3. However, there are several other schemes proposed in recent years that do not fall into any of the categories discussed above. In the direction of enhancing robustness by bias reduction, the work [122] tries to remove hypothesis-only bias for NLI datasets by using adversarial classifiers to detect bias in the sentence representation. They demonstrated that the larger the sentence embeddings, the harder it is to remove the bias and requires more adversarial classifiers. They tested models with 1 and 20 classifiers where 8 out of 13 datasets performed better with 20 classifiers and for 3 of them 1 and 20 gave the same performance.

In another line of defending APIs from adversarial attacks the work [40] showed that hosted BERT-based APIs are vulnerable to theft and users can query the API for a dataset and train a BERT model to replicate the API. The replicated model can then be used for adversarial example transfer. They suggested a parameter-based defense strategy by using a temperature parameter in softmax to smooth the output prediction probabilities. They further add perturbation noise with variance sigma to the output probabilities where the larger the variance stronger the defense.

In the direction of creating various adversarial examples for adversarial training, the work [38] proposed variable length adversarial attack in contrast to an existing method which focuses on fixed length. This is achieved by using a special "BLK" token during fine-tuning and then using 3 atomic operations addition, deletion & replacement to create adversarial examples. They show that this method successfully attacks the models in NLU and NAT tasks and demonstrated its use for creating augmented data for adversarial training. In the same line, the authors in [31] presented backdoor attacks as adversarial attacks during training of the model for NLG models. They proposed post-hoc defense against the attacks by using token removal and token substitution on a sentence and corpus level.

Taking VAT in different directions, [176] proposed a novel strategy, Stackelberg Adversarial Training (SALT) which employs a Stackelberg game strategy. There's a leader which optimizes the model and a follower which optimizes the adversary. In this Stackelberg strategy, the leader is advantageous knowing the follower's strategy and this information

is captured in Stackelberg gradient. They find the equilibrium between the leader and follower using an unrolled optimization approach.

Another work in the direction of robustness enhancement, proposed in [53] introduced Robust Encodings (RobEn), which is a simple framework that guarantees robustness, without making any changes to model architecture. The core component of RobEn is an encoding function, which maps sentences to a smaller, discrete space of encodings on a token level. They attempt to cluster all possible adversarial typos into a single cluster using graph-based agglomerative clustering and try to balance between having too many words in a cluster versus a single word in a cluster. In the same line, for languages other than English, [73] proposed AdvGraph to enhance the adversarial robustness of Chinese NLP models with modified embeddings. Due to the inherent complexity in Chinese language, the existing adversarial defense models are difficult to be extended for the Chinese language, hence they propose to capture the similarity in words using graphs. They constructed undirected adversarial graph based on the glyph and phonetic similarity of Chinese characters and learned the representations through graph embeddings to be used with semantic embeddings to be used for other downstream tasks.

There are various metrics used in the literature for evaluating the robustness of the models against adversarial attacks. Evaluating adversarial robustness is equivalent to evaluating the performance of a model in the presence of adversarial attacks, before and after the implementation of defense mechanism. Hence, some of these metrics are standard performance metrics widely used in machine learning literature. Other variants of evaluation metrics are proposed to measure the performance of certifiably robust models. The next section describes these evaluation metrics used in adversarial defense literature.

8 METRICS FOR EVALUATION

There are various evaluation metrics that are extensively used in literature for evaluating the proposed defense methods. Majorly these metrics are performance evaluation metrics for the machine learning model which is required to be defended. Adversarial defense methods are evaluated with the performance of the model after the defense method is implemented or run-time evaluation in case of methods that aim towards optimizing a lower/upper bound. Hence, some of these metrics are accuracy, error, loss analysis, measurement of the success of adversarial attacks, similarity with ground truth in the case of language generation etc. In this section, some of these commonly used metrics are described in detail.

- **Prediction accuracy (Conventional Accuracy)**: Adversarial defense methods for text classification models, such as sentiment classification, Natural Language Inferencing tasks, are evaluated on the basis of the prediction accuracy after implementation of defense algorithm. The prediction accuracy after defense is compared with prediction accuracy after attack, and if there is a surge in accuracy, the defense method is considered to be successful. It is defined as the fraction of test set that is correctly classified [135]. Conventional accuracy is a standard metric that is used to evaluate any deep learning system and it can be used to evaluate any defense method.
$$\frac{\sum_{t=1}^{T} CorrecClass(X_t, L, \epsilon)}{T}$$
Here, $CorrecClass(X_t, L, \epsilon)$ gives 1, if test sample, X_t is correctly classified for test data T.
- **Loss function analysis**: The negative log-likelihood (loss function) is tested over its rate for adversarial training as regularization, and virtual adversarial training-based methods. It can indicate lower error rate and reduced overfitting in adversarial training based regularization.

- **Error analysis**: Adversarial defense methods are also evaluated on the error rate in the prediction of the model. The error rate is compared with an adversarial attack before and after the implementation of defense schemes. A lesser error rate after defense method, entails its successful defense scheme. Error analysis is a standard metric that is used to evaluate any deep learning system for prediction and it can be used to evaluate any defense method in literature.
- **Embedding testing**: Embedding test is done for evaluating the embeddings generated for adversarial training. Similarity metrics such as Edit distance, Jaccard similarity coefficient, and semantic similarity metrics are used to evaluate the utility of the adversarial samples generated by finding their similarity with the original input samples.
- **Human Evaluation**: To measure the utility of the adversarial samples for adversarial training, human evaluation is also performed in literature [34]. Human annotators are asked to judge the adversarial examples in terms of their naturalness by presenting both original and adversarial examples. The good quality adversarial examples are used in adversarial training and further robustness of the deep neural network is evaluated. Adversarial attacks and defense methods are closely associated, hence weaker the adversarial attack, the stronger the defense strategy.
- **Attack Success Rate (ASR)**: ASR [145] is a measure of success of the adversarial samples created for the potential adversarial attack. This metric is used to measure the effectiveness of the adversarial attack after any of the defense scheme is implemented. The attack success rate is measured before and after defending the model and a drop in its value will imply a more robust model. It is defined as:

$$ASR = \frac{N_{successful}}{N_{Total}} * 100$$

  Where, $N_{successful}$ is the number of adversarial samples that were able to successfully fail the model, and N_{Total} is the total number of adversarial samples generated.
- **BLEU**: Adversarial defense schemes for Natural language generation models utilize performance metrics such as BLEU score [98] for the evaluation of their proposed method. BLEU score is measured using n-gram to evaluate the quality of the generated natural language by comparing it with ground truth. It is defined as:

$$p_n = \frac{\sum_{C \in \{Cand\}} \sum_{gram-n \in C} Count_{clip}(gram - n)}{\sum_{C' \in \{Cand\}} \sum_{gram-n' \in C'} Count(gram - n')}$$

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{\frac{1-r}{c}}, & c \leq r \end{cases}$$

$$BLEU = BP.exp^{\sum_{i=1}^{N} W_n log(p_n)}$$

  Where (p_n) is the n-gram modified precision score, BP is brevity penalty used for longer candidate summaries and for spurious words in it, c is the length of the candidate summary, and r is the length of the reference summary.
- **Number of Queries**: This denotes average number of times attacker queries the model. The higher the average number of queries made by an attacker, more difficult is to fail a defense mechanism [81]. It can be used for evaluating any adversarially trained defense model using adversarial instances or by controlling the perturbations.

- **Precision on certified examples**: Precision on certified examples [67], which measures the number of correct predictions exclusively on examples that are certified robust for a given prediction robustness threshold. It is defined as:

$$Precision = \frac{\sum_{t=1}^{T}(isCorrect(X_t)\ \&\ robustSize(p_t, \epsilon, \delta, L) \geq Threshold}{\sum_{t=1}^{T} robustSize(p_t, \epsilon, \delta, L) \geq Threshold}$$

  Where, $isCorrect(X_t)$ gives a value of 1 if the input sample X_t is correctly classified, and $robustSize$ gives the certified robustness value for the bound L.
- **F1 Score**: This is a metric that combines precision and recalls both, to evaluate the overall performance of the model. It is used for comparison of model's performance before and after defense mechanism implementation. F1 score is a standard metric used for evaluating the performance of deep neural networks and it can be used to evaluate any defense method. It is defined as:

$$F1\ Score = \frac{2 * (Precision * Recall)}{(Precision + Recall)}$$

- **Certified Radius**: In certification based adversarial defense methods, robust radius is the largest radius centered around input sample X_t, for which the classifier does not change its value for its corresponding perturbed sample X_t^{adv}. However, calculating robust radius of a deep neural network is a NP-hard problem [163]. Hence, certification based methods are tested for their certified radius for different norms of perturbations for targeted model on parameters such as minimum radius, average radius, and time taken to obtain it. Certified radius is a lower bound to the robust radius and leads to a guaranteed upper bound of the robust classification error.
- **Certificate Ratio (CR)**: This metric is used in certification based defense schemes. It is the fraction of testing samples, that satisfies the certification criteria after prediction [135]. It is defined as:

$$CR = \frac{\sum_{t=1}^{T} CertifiedCheck(X_t, L, \epsilon)}{T}$$

  Here, $CertifiedCheck(X_t, L, \epsilon)$ gives 1, if the fraction of the test data is certified robust.
- **Certified Robustness**: This metric is used in certification-based defense schemes. Certified Robustness [70] for a particular X_t is the maximum value ρ for which it is certified that classifier will return the correct label where X_t^{adv} is its corresponding perturbed sample, such that $||X_t - X_t^{adv}|| \leq \rho$.
- **Median Certified Robustness**: This metric is also used in certification-based defense schemes. The Median Certified Robustness [70, 162] on a dataset is the median value of the certified robustness across the dataset. It is the maximum value ρ for which the classifier can guarantee robustness for at least 50% samples in the dataset. In other words, we can certify the classifications of over 50% samples to be robust to any perturbation within ρ.
- **Certified accuracy**: This is the metric used for evaluating the certifiable robust models. Certified accuracy [67, 135] is the percentage of correct test samples for a certified robust model for the given perturbation. It denotes the fraction of testing set, on which a certified model's predictions are both correct and certified robust for a given prediction robustness threshold. It is defined as:

$$Certified\ Accuracy = \frac{\sum_{t=1}^{T}(isCorrect(X_t)\ \&\ robustSize(scores, \epsilon, \delta, L) \geq Threshold}{T}$$

  Where $robustSize(scores, \epsilon, \delta, L)$ is the certified robustness size for the bound L and $isCorrect(X_t)$ give a value of 1 if the input sample X_t is correctly classified for test data T.

- **Conditional Accuracy**: This is the metric used for evaluating the certifiable robust models. Conditional Accuracy is proposed by [135], evaluating the classification accuracy of both a clean sample X_t and its corresponding adversarial sample X_t^{adv} withing a bound L. It checks when X_t is certified within bound L, whether X_t^{adv} is also classifying correctly. It is defined as:

$$Conditional\ Accuracy = \frac{\sum_{t=1}^{T}(Certified(X_t, L, \epsilon)\ \&\ corrClass(X_t^{adv}, L, \epsilon)}{\sum_{t=1}^{T}(Certified(X_t, L, \epsilon)}$$

  Where, $Certified(X_t, L, \epsilon)$ gives 1, when clean input sample X_t is successfully certified and $corrClass(X_t^{adv}, L, \epsilon)$ gives 1 when its perturbed input X_t^{adv} is also correctly classified.
- **CLEVER score**: Cross Lipschitz Extreme Value for nEtwork Robustness (CLEVER) score is proposed by [143] is a novel robustness evaluation metric. It is attack-independent and can be applied to any arbitrary neural network classifier and scales to large networks. CLEVER metric is an estimation of local Lipschitz constant which represents “lower bound of the robustness in input data" or minimum amount of perturbation required to a natural sample to fail a classifier. The increased clever score indicates that the network is indeed made more resilient to adversarial perturbations after any defense mechanism is used.

9 ADVERSARIAL DATASETS AND FRAMEWORKS

There are several dataset in NLP which are proposed for adversarial evaluation. One such dataset is DailyDialog++ [115], which is an extension of DailyDialog dataset [80] and adversarial dialogue generation dataset. DailyDialog++ contains, 5 additional relevant and adversarially irrelevant responses, for 11k context conversation derived from DailyDialog. This dataset is used for evaluating robustness of dialogue generation models with adversarial examples in the dataset against adversarial attacks. Further these adversarial examples are used in adversarial training to imporve the performance of dialogue generation models. Another work is ANLI [94], which proposed adversarial dataset for natural language inference systems. ANLI composed of adversarial examples for NLI collected in 3 iterative rounds having human and machine in loop. ANLI consist of $103k$ examples of sentences, starting with short multi-sentence passages from Wikipedia and having annotators writing adversarial hypothesis. In this process, they tested these samples with state-of-the-art NLI models and got then verified by human annotators hence proposing human-and-model-in-the-loop enabled training (HAMLET) scheme for data collection. The adversarial examples collected in this dataset are used in adversarial training to improve the performance of Natural Language Inferencing models.

In the same line, authors in [76] proposed a novel large scale dataset adversarial VQA for visual question answering task using the HAMLET scheme proposed in [94]. In this work they presented an image to an annotator and ask them to write a tricky question that could fool a model. Hence they iteratively collected $243.0K$ questions for $37.9K$ images by having humans and models competing in the loop in 3 rounds. This dataset is used in evaluating the robustness of SOTA VQA models. Authors have further used this dataset for data augmenting, for the purpose of adversarial training and demonstrated a higher performance on robust VQA benchmarks. For the purpose of evaluating adversarial examples for question answering task, authors in [50] proposed, Adversarial SQuAD dataset, that contained adversarially inserted sentences. These sentences are automatically generated in a concatenative manner, without changing the meaning of the paragraph or question. Fake answers to these questions are also generated with same POS type. This dataset is used for evaluating the performance of various state-of-the-art question answering models. The best performing model with adversarial examples is analysed for the features causing the high performance. Based on this analysis, additional features were incorporated in the model, separately and in combination, to enhance its performance both with and

without adversarial examples. Authors further used this dataset for adversarial training of modified model to validate its stability. In the same line, [129] proposed a question-answering test bed Quizbowl, using a Human-In-the-Loop framework. In this work, human authors are asked to write adversarial questions which are designed to fail state-of-the-art question answering models appearing ordinary to human. This test dataset is used for adversarial evaluation of SOTA question answering models.

In another work, [169] adversarial dataset, namely, Paraphrase Adversaries from Word Scrambling (PAWS), for paraphrase detection is proposed. PAWS is generated from sentence in Quora and Wikipedia, where adversarial samples are generated using language model based controlled word swapping and back translations. PAWS dataset is used for adversarial evaluation of paraphrase detection models and measured sensitivity of these models on word order and syntactic structure. Authors further used this dataset for adversarial training of state-of-the-art paraphrase detection models. In the direction of perturbation identification [147] proposed a dataset Text Classification and Attack Benchmark (TCAB), which was created for the purpose of detecting and labeling the textual perturbation in the input text. TCAB is an extensive dataset containing 1.5 million attack instances, generated using 12 attacks from the toolkits [91, 161] which are targeting 3 classifiers, trained on 6 domain datasets of sentiment and abuse classification. There is a total number of 216 attacks taken into consideration in TCAB dataset which are a combination of different type of attacks. This dataset consists of a total of $1,53,9881$ adversarial examples along with the clean instances taken from the original datasets. They further use text, language, classifier properties to detect the perturbation and type of the attack in the input text. Table 9 shows the summary of the publicly available adversarial datasets for various NLP tasks. It describes the task for which the dataset is proposed, its statistics and methods used in data collection and annotation.

Dataset	Task	Statistics	Method
DailyDialog ++ [115]	Dialogue Generation	$11k$ context, 5 responses	Human Annotators
ANLI [94]	NLI systems	$103k$ sentences	Human & machine in loop
Adversarial VQA [76]	Visual Question Answering	$243.0k$ Q & $37.9k$ images	Human & machine in loop
Adversarial Squad [50]	Question Answering	$107,785$ Q	Machine generated
Quizbowl [129]	Question Answering	1213 Q-A pairs	Human & machine in loop
PAWS [169]	Paraphrase detection	$108,463$ paraphrase pairs	Machine generated
TCAB [147]	Sentiment, Abuse classification	$1.5m$ Adversarial ins.	Machine generated

Table 9. Summary of adversarial datasets in NLP, here NLI = Natural Language Inference, Q= Questions, A= Answers, ins= instances

There are papers in the literature that proposed python frameworks for a complete adversarial evaluation for several NLP tasks with various attack algorithms. One such work is TextAttack [91] which is a python framework for end-to-end adversarial evaluation with 16 different adversarial attack methods. It consists of a task-specific goal function, data augmentation schemes along with perturbation constraints that validate the perturbation with original inputs, and a repetitive model querying search system. It facilitates the user to benchmark existing attacks, and create novel attack schemes by using new and existing components and evaluating them. Improving the shortcomings of TextAttack framework, Open attack [161] is proposed, which included 15 different types of attacks such as sentence, word, character level. It also supports Chinease in addition to English language models and supports multi-process running of attack models to improve attack efficiency. In the same line another evaluation framework [151] “Elephant in the room" is proposed, which consists of a combination of automatic evaluation metrics and human judgments. Targeting the sentiment classification task, it included crowd-sourced human judgments, for judging the naturalness, preservation of original label, and comparing similarity on a text similarity metric.

10 RECOMMENDATIONS FOR FUTURE WORK

In this paper, an exhaustive survey of the methods proposed in the literature to defend neural networks from adversarial attacks and to enhance their robustness is presented. We proposed a novel taxonomy for adversarial defense mechanisms for various tasks in NLP. Methods for adversarial defense in NLP are broadly divided into three categories, (i) methods based on adversarial training, (ii) methods based on perturbation detection (iii) methods providing a certificate for robustness. Another part of methods which do not follow any of the above mentioned schemes is categorized as miscellaneous. While there is an ample amount of work proposed in this direction for tasks in NLP, there are still various gaps remaining which should be looked at as potential future directions in this area.

- **Larger part of work based on adversarial training**: A large portion of work in adversarial defenses for NLP revolves around adversarial training and data augmentation. Despite having a plethora of work in this direction, there is a large part of adversarial defense methods which are oriented towards augmenting the data for generating adversarial examples for adversarial training. Adversarial training is undoubtedly a successful defense scheme for adversarial attacks but it lacks generality for a more practical purpose. It makes the model highly robust for a certain kind of attack but it still makes it vulnerable to the type of examples the model has not seen. Hence, the other methods for adversarial defenses should also be explored.
- **Hand-crafted generation of adversaries**: As a future work recommendation, more attention should be given to the automatic generation of adversarial examples. Most of the methods based on adversarial training with data augmentation rely on hand-crafted adversarial examples. There are methods that replace words with their synonyms, or adjacent words, flip the character or concatenate words at the end of sentences. Despite being highly efficient, these examples are devised by humans rather than having an automatic generation of adversarial examples and have their own limitations. Hence, more efforts can be put in the direction of the automatic generation of adversarial examples.
- **Interpretability of perturbations in text data**: Adversarial examples should be less human perceptible/natural looking to make the model robust in a practical attack scenario. In contrast with adversarial defenses for computer vision based methods and examples generated on images, examples on text for NLP tasks are difficult to generate because of their discrete nature. Modifying pixels in images for generating adversarial examples are less perceptible to human than modifying a character or word in an input text. Hence to make to defense method more useful in a practical scenario, more efforts should be put in this direction.
- **Exact robust certificate calculation**: In existing literature only the upper and lower bound on the certificate could be calculated, rather than the exact robustness certificate. While adversarial training based defense methods provided a sufficient exposure towards achieving robust neural networks, progressively novel adversarial attacks kept rolling in. A new set of methods to achieve robustness were proposed in the direction of providing a certificate of robustness for a neural network, attempting to put an end to this race. Certification based adversarial defense methods, definitely provides a more generalization for the neural network for a task but certification does not make a sufficient property of a model for achieving robustness. Finding an exact robustness certificate to the set of input is a non-convex optimization problem and is inefficient to solve. However, in literature authors have relaxed this problem to convex optimization by finding an upper or lower bound to the robustness certificate. Despite the efforts towards finding a certified robustness convex optimization can lead to lossy results and there is a scope of finding a tighter bound. Hence future efforts in this direction can be made to improve the tightness of existing robustness certificates.

- **Scalability of certification based robustness**: Certification methods, do not scale to large and practical networks used in solving modern machine learning problems. The current certification based robustness method in literature, is implemented on theoretical models on a small scale. They are not scalable to larger and deeper networks for practical purposes. Hence, in future attempts should be made in this direction.
- **Generalization of adversarial training**: The current state of the art methods based on adversarial training in NLP are designed in a task specific manner. There is a lack of generality in adversarial example generation schemes which could be used for multiple NLP tasks effectively. Therefore, steps can be taken in this direction in the future.
- **More explainable models in case of inserting perturbations in the loss function**: There is a lack of explainability and transparency in the regularization based defense methods. The loss function contains the term responsible for introducing perturbations at the training time. However, there is no explanation for these methods for their correctness or high accuracy.
- **Interpretable adversarial training**: While adversarial training based method covers a larger portion of adversarial defense literature in NLP, these models are hardly interpretable in terms of adversarial instances. Adversarial examples are provably helping these models become more robust toward adversarial attacks, but there are questions such as, “how they are improving the performance", “is this list of adversarial instances exhaustive or could there be more such instances", “is the model robust towards some specific type of attacks or they are able to defend from any type of attacks", are still need to be answered.
- **Attack agnostic perturbation detection**: A better part of the literature in perturbation detection methods works on the prior assumption of spelling-based attacks or synonym substitution based attacks. However, adversarial perturbation in textual data could be caused by multiple types of attacks, individually or in combination. Hence, there is a requirement for perturbation detection schemes that attack agnostic and present a general framework for perturbation detection.
- **Novel methods to identify the existing perturbations in the input**: A large part of perturbation detection schemes depends upon spell checking methods and methods which enumerate or cluster synonyms. In a practical scenario, it is highly inefficient to compute and enumerate synonyms of words for perturbation recognition. Hence, there is a requirement for novel and innovative methods for identifying the perturbations in the input text which do not involve traditional methods such as spell checking, synonym mapping, or their other variants.
- **Better coverage for NLP applications**: The proposed adversarial defense methods evaluate their method on certain NLP tasks to validate strength of their method. In addition to being limited to certain type adversarial attacks, a large part of defense methods demonstrate their algorithm on different types of text classification tasks. These tasks include sentiment, hate, news, topic, abuse, malware classification type of problem which comes under the umbrella usecase of text classification. Text classification is one of the Natural Language Understanding (NLU) task, while there are other NLU tasks such as named entity recognition, machine translation, automatic reasoning etc. having lesser coverage. Moreover, a very limited number of defense schemes have demonstrated their methods on Natural Language Generation (NLG) use cases such as summary generation, question answering. Hence, there is a requirement for adversarial defense methods covering the other NLP applications.
- **Better evaluation metrics**: The current evaluation of robustness against adversarial attacks for NLP models is based on the performance metrics of the actual model, i.e. accuracy, precision-recall, error-analysis, etc. Hence, there is a requirement for novel evaluation metrics that could measure the robustness and ability to defend against adversarial attacks. There is also a requirement for sensitivity metrics for machine learning models, for

measuring their sensitivity towards adversarial examples. There also ain't enough ways to evaluate defense mechanisms themselves along the lines of perceptibility and naturalness. Hence, in future, more evaluation metrics can be brought along this line.

11 CONCLUSION

In this paper, a survey is presented for adversarial defense methods for various tasks in NLP. We proposed a novel taxonomy for adversarial defenses in NLP covering a wide range of recently proposed papers. This survey tries to fulfill the gaps in existing surveys where adversarial attack schemes were more focused. However, in recent years, numerous methods are proposed for defending neural networks with adversarial attacks and enhancing their robustness. Coming up with novel defense schemes for advanced NLP systems is as important as coming up with novel attacks to make these neural networks robust and safe for practical purposes. This survey also covers various adversarial datasets, frameworks proposed in recent times for efficient adversarial evaluation of the SOTA models, and evaluation metrics to quantify their robustness. Moreover, it highlights various recommendations for future work considering the limitations and gaps in the existing literature on adversarial defenses. This survey, therefore, provides a strong basis and motivation for future research in developing robust and safe neural networks in NLP tasks.

A VARIOUS NATURAL LANGUAGE PROCESSING TASKS

In this section various NLP tasks are described in detail, on which adversarial defense schemes are demonstrated in literature. These tasks include applications of both Natural Language Understanding (NLU) and Natural Language Generation (NLG).

A.1 Text classification

Text classification is an essential NLP task that involves classifying text data into multiple categories. It encompasses a vast array of problems, including but not limited to sentiment analysis, natural language inference, malware detection, spam filtering, and topic labeling. In the text classification process, a contextual representation of the input text is generated, followed by the selection of an appropriate classifier [1, 63, 79]. This process may also involve preprocessing and intermediate steps such as tokenization, lemmatization, stemming, and dimensionality reduction.

A.2 Named Entity Recognition

Named Entity Recognition (NER) is a NLP task that involves identifying proper nouns or rigid entities in text such as organizations, persons, locations, and quantities [75, 126, 152]. This is an important task for various applications like question answering, information retrieval, relation extraction, text summarization, and machine translation. There are four major categories of NER techniques, including Rule-based NER that uses handcrafted rules, unsupervised learning approaches that rely on clustering based on contextual similarity, supervised learning approaches that employ feature engineering with conventional classification schemes such as HMM, SVM, or other classifiers, and Deep learning-based approaches [75, 126, 152].

A.3 Part-of-Speech tagging

Part-of-speech (POS) tagging is an NLP task that involves assigning a specific category to each token in a given text, such as verb, noun, or adjective. POS tagging methods are classified into four categories: rule-based, stochastic, transformation-based, and HMM-based [55]. Rule-based taggers use a set of predefined rules to tag words. Stochastic methods, on

the other hand, are probability-based and use frequency information to disambiguate the tags. Transformation-based taggers combine both rule-based and stochastic methods, where a small set of rules are learned from the data and tagging is transformed in each cycle. HMM-based taggers find the most probable sequence of POS tags for a given input sequence, modeled using Hidden Markov Model (HMM) [14, 54].

A.4 Machine comprehension and Question answering

Question answering is a task that has been extensively studied in the NLP literature, with the goal of building systems that can automatically generate answers to questions posed in a given context. This task has numerous applications, including the development of chatbots and dialogue generation systems. In order to train a system to perform this task, a neural network is trained on a large dataset of contexts, questions, and their respective answers, learning the relationships among them. The SQuAD dataset [105], for example, has been proposed for this purpose, containing a large number of context paragraphs, questions, and answers.

A.5 Automatic summarization

Automatic text summarization is a challenging NLP task that involves creating a shorter version of a larger text document. The task requires the selection of essential information from the entire text and condensing it using the sentences available in the document or a different set of vocabulary. There are two ways to perform text summarization: extractive summarization and abstractive summarization, which are determined by the available training data and the desired output. Extractive summarization involves selecting keywords, phrases, and lines from the document and combining them to create a summary, making it useful when the available training data is limited [41, 112]. On the other hand, abstractive summarization is a data-driven approach that involves training a machine learning model to create a summary using the learned language in the available data [41, 93]. Abstractive summarization requires a large amount of data but produces a high-quality summary of the text document.

A.6 Machine translation

The task of machine translation involves the conversion of text from a source language to a target language, which can be either in the form of sentences or documents. Based on the availability of a large training corpus and similarities between languages, MT systems can be modeled as either monolingual or multilingual [15, 106, 153]. However, the task becomes more complex when dealing with diverse domains such as legal, medical, and cultural texts. Additionally, the mode of text, whether formal written [7] or informal spoken [117], poses a challenge for the task. Moreover, machine translation is susceptible to hallucinations [107] and can be vulnerable to adversarial attacks.

A.7 Dialogue generation

Automatic dialogue generation is a crucial NLP task that focuses on building conversational systems capable of generating responses automatically, with the goal of creating a human-like conversation system. These systems can be either task-oriented, designed to operate within a specific domain such as transportation, restaurants, or shopping, or open dialogue systems for open-ended conversations, such as everyday conversation between two people. Dialogue generation systems have evolved over three phases, including rule-based systems, retrieval-based systems , and neural generative conversation systems [125]. The quality of the generated response is dependent on the system's intelligence, as it should be consistent with the previous statement and align with the context of the conversation and targeted domain (if applicable).

A.8 Question generation

Automatic question generation is a NLP task that is designed to facilitate educational assessment by generating questions automatically from a given text. This task is important as it helps to reduce manual labor and time. The questions generated by these systems should be well-phrased and must have answers that can be found in the given piece of text. These questions can be of two types: subjective or objective [23]. Question generation systems have applications in a variety of fields such as Massive Open Online Courses (MOOC), healthcare systems, chatbots, and search engines [95].

B ILLUSTRATIONS OF SOME ADVERSARIAL DEFENSE METHODS IN NLP

In this section some of the popular adversarial defense schemes in NLP are described with illustrations, across various categories proposed in the paper.

B.1 AdvEntuRe: Adversarial Training for Textual Entailment with Knowledge-Guided Examples

In the direction of GAN based adversarial training, the work proposed in "AdvEntuRe: Adversarial Training for Textual Entailment with Knowledge-Guided Examples" [56] uses adversarial training with GANs for the textual entailment task. The generator and discriminator are trained in an end-to-end manner, where generator (seq2seq) is trained for generating adversarial examples using external knowledge or handwritten rules. Discriminator is trained in the same manner to learn the textual entailment for the generated samples. Figure 2 presents the overall pipeline proposed in this work for GAN-based adversarial training.

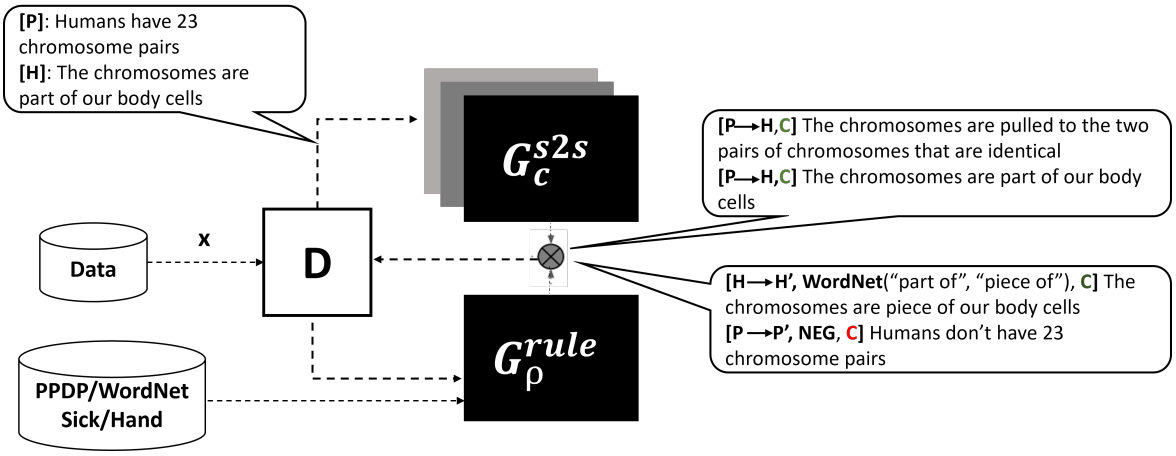

Fig. 2. Overall pipeline of Adventure- Knowledge guided textual entailment [56]

B.2 Natural Language Adversarial Defense through Synonym Encoding

In the direction of perturbation control based adversarial defense methods, "Natural Language Adversarial Defense through Synonym Encoding" [139] proposed perturbation identification scheme. Perturbations related to word modifications which included insertion, deletion, substitution or swapping of words are identified in several ways. In this work, the authors proposed defense mechanism, against synonym substitution, calling it "Synonym Encoding Method"(SEM). They essentially clustered all the synonyms in embedding space with their euclidean distances and then encoder is

layered before input to train the model. Encoder is responsible for identifying all the synonym substitution-based attacks in the model and maps all the synonyms to a unique encoding without adding extra data for training. Figure 3 demonstrates the Synonym Encoding Method proposed described above.

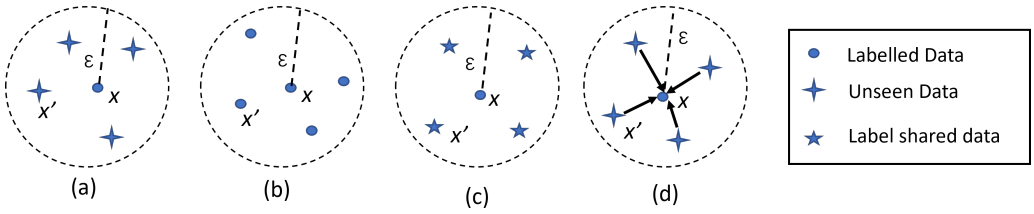

Fig. 3. The neighborhood of a data point x in the input space in [139]. (a) Normal training has unseen data points x, which leads to wrong classification. (b) Training with infinite labeled data to overcome all possible adversarial attacks. (c) Training with data with shared labels, in which all the neighboring points share the labels. (d) Mapping the neighboring points to center x to remove adversaries.

B.3 Interpretable Adversarial Perturbation in Input Embedding Space for Text

The proposed work "Interpretable Adversarial Perturbation in Input Embedding Space for Text" [119], under perturbation direction control category alters the direction of the perturbations towards the cleaner text input limiting the adversarial space. Along this line, this work proposed an interpretable adversarial training method by restricting the direction of adversarial samples. The direction of perturbation is restricted to the words in the existing vocabulary so that perturbations could be interpreted even after adversarial training. Figure 4 shows the mechanism to restrict the direction of the perturbations.

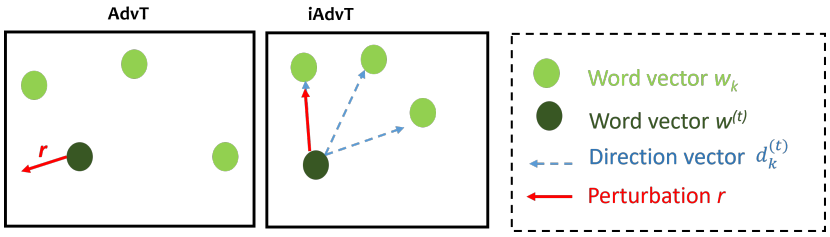

Fig. 4. Restricting the direction of the perturbations proposed in [119]. The proposed method (iAdvT) restricts the perturbations in the direction of input word embeddings, while the previous methods (AdvT) lets them choose any direction

B.4 SAFER: A Structure-free Approach for Certified Robustness to Adversarial Word Substitutions

In the work, "SAFER: A Structure-free Approach for Certified Robustness to Adversarial Word Substitutions" [157] in the direction of robustness by certification, proposed structure-free certified robust models which can be applied to any arbitrary model. This method overcomes the limitations of IBP based method in which they are not applicable to character level and sub-word level models. They prepared a perturbation set of words using synonym sets, top-K nearest neighbors under the cosine similarity of GLOVE vectors, where K is a hyperparameter that controls the size of the perturbation set. They further generated sentence perturbations using word perturbations and trained a classifier

with robust certification. Figure 5 presents the overall pipeline to achieve certified robustness. In the context of IBP methods,

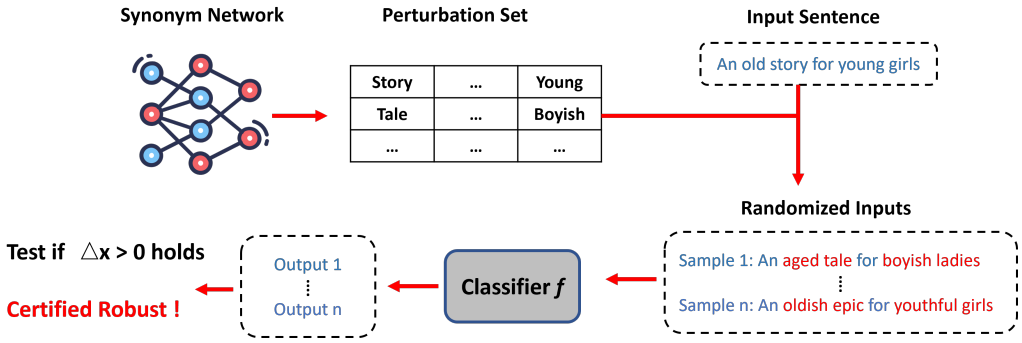

Fig. 5. Pipeline for certified robustness approach proposed in [157]

B.5 Certified Robustness to Word Substitution Attack with Differential Privacy

The work "Certified Robustness to Word Substitution Attack with Differential Privacy" [135] proposed in the direction of robustness by certification, provided a certificate of robustness with the idea of differential privacy in the input data. They implemented differential privacy in the textual data by treating a sentence as a database and words as an individual records. If a predictive model satisfies a certain threshold (epsilon-DP) for a perturbed input, its input should be the same as the clean data. Hence providing a certification of robustness against L-adversary word substitution attacks. Figure 6 demonstrates the certified robustness framework proposed in [135].

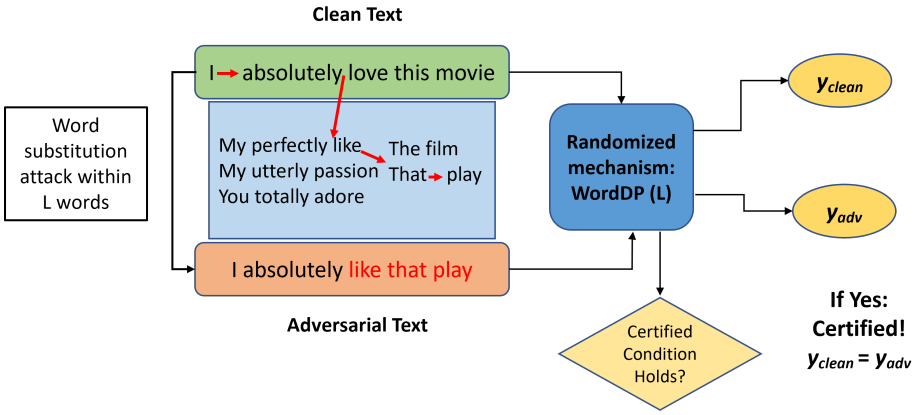

Fig. 6. Word substitution attack and certified robustness via wordDP proposed in [135]

B.6 Adversarial NLI: A New Benchmark for Natural Language Understanding

In the direction of adversarial datasets and framework, ANLI [94], proposed adversarial dataset for natural language inference systems. ANLI composed of adversarial examples for NLI collected in 3 iterative rounds having human

and machine in loop. ANLI consist of 103k examples of sentences, starting with short multi-sentence passages from Wikipedia and having annotators writing adversarial hypothesis. In this process, they tested these samples with state-of-the-art NLI models and got then verified by human annotators hence proposing human-and-model-in-the-loop enabled training (HAMLET) scheme for data collection. The adversarial examples collected in this dataset are used in adversarial training to improve the performance of Natural Language Inferencing models. Figure 7 depicts the dataset collection framework for ANLI dataset proposed in [94].

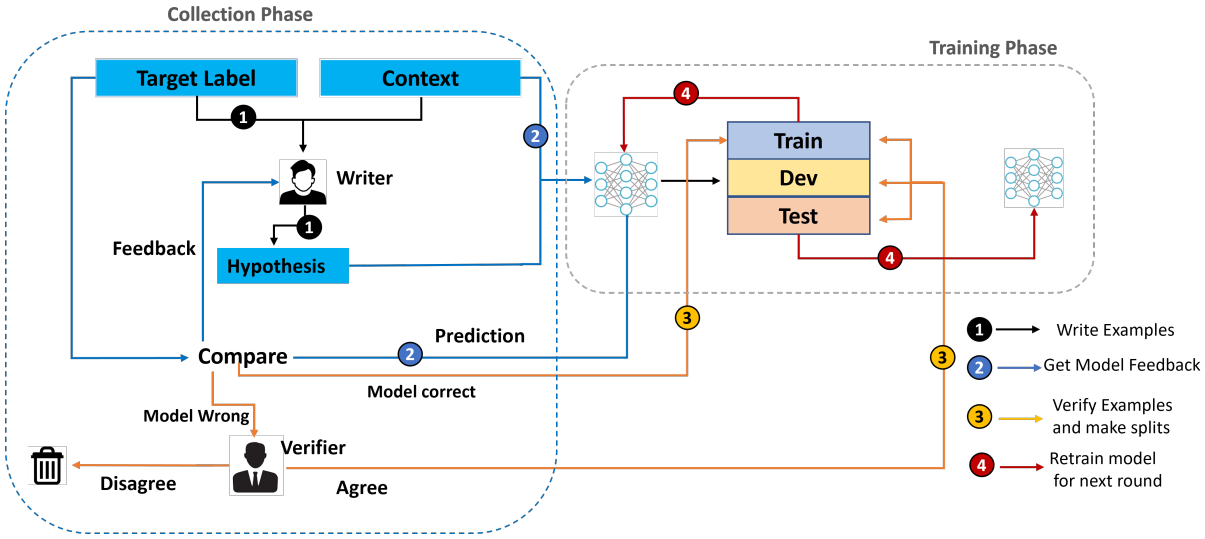

Fig. 7. Adversarial NLI data collection framework with Human-and-model-in-the-Loop proposed in [94]

REFERENCES

[1] Charu C Aggarwal and ChengXiang Zhai. 2012. A survey of text classification algorithms. In *Mining text data*. Springer, 163–222.

[2] Naveed Akhtar and Ajmal Mian. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access* 6 (2018), 14410–14430.

[3] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani B. Srivastava, and Kai-Wei Chang. 2018. Generating Natural Language Adversarial Examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, 2890–2896. https://doi.org/10.18653/v1/d18-1316

[4] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*. PMLR, 274–283.

[5] Vincent Ballet, Xavier Renard, Jonathan Aigrain, Thibault Laugel, Pascal Frossard, and Marcin Detyniecki. 2019. Imperceptible adversarial attacks on tabular data. *arXiv preprint arXiv:1911.03274* (2019).

[6] Rongzhou Bao, Jiayi Wang, and Hai Zhao. 2021. Defending Pre-trained Language Models from Adversarial Word Substitutions Without Performance Sacrifice. *arXiv preprint arXiv:2105.14553* (2021).

[7] Loic Barrault, Ondrej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussa, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Tom Kocmi, Andre Martins, Makoto Morishita, and Christof Monz (Eds.). 2021. *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics, Online. https://aclanthology.org/2021.wmt-1.0

[8] Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and Natural Noise Both Break Neural Machine Translation. arXiv:1711.02173 [cs.CL]

[9] Petr Bělohlávek. 2017. Using adversarial examples in natural language processing. (2017).

[10] Rasika Bhalerao, Mohammad Al-Rubaie, Anand Bhaskar, and Igor Markov. 2022. Data-Driven Mitigation of Adversarial Text Perturbation. *arXiv preprint arXiv:2202.09483* (2022).

[11] Siddhant Bhambri, Sumanyu Muku, Avinash Tulasi, and Arun Balaji Buduru. 2019. A survey of black-box adversarial attacks on computer vision models. *arXiv preprint arXiv:1912.01667* (2019).

[12] Gregory Bonaert, Dimitar I Dimitrov, Maximilian Baader, and Martin Vechev. 2021. Fast and precise certification of transformers. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*. 466–481.

[13] Nicholas Boucher, Ilia Shumailov, Ross Anderson, and Nicolas Papernot. 2022. Bad characters: Imperceptible nlp attacks. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1987–2004.

[14] T. Brants. 2000. TnT - A Statistical Part-of-Speech Tagger. *ArXiv* cs.CL/0003055 (2000).

[15] Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906* (2017).

[16] Vanessa Buhrmester, David Münch, and Michael Arens. 2021. Analysis of explainers of black box deep neural networks for computer vision: A survey. *Machine Learning and Knowledge Extraction* 3, 4 (2021), 966–989.

[17] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2018. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069* (2018).

[18] Luoxin Chen, Weitong Ruan, Xinyue Liu, and Jianhua Lu. 2020. SeqVAT: Virtual adversarial training for semi-supervised sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 8801–8811.

[19] Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust Neural Machine Translation with Doubly Adversarial Inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 4324–4333. https://doi.org/10.18653/v1/P19-1425

[20] Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020. Advaug: Robust adversarial augmentation for neural machine translation. *arXiv preprint arXiv:2006.11834* (2020).

[21] Siddhartha Chib and Edward Greenberg. 1995. Understanding the metropolis-hastings algorithm. *The american statistician* 49, 4 (1995), 327–335.

[22] Maximin Coavoux, Shashi Narayan, and Shay B Cohen. 2018. Privacy-preserving neural representations of text. *arXiv preprint arXiv:1808.09408* (2018).

[23] Bidyut Das, Mukta Majumder, Santanu Phadikar, and Arif Ahmed Sekh. 2021. Automatic question generation and answer assessment: a survey. *Research and Practice in Technology Enhanced Learning* 16, 1 (2021), 1–15.

[24] Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. *arXiv preprint arXiv:1908.06083* (2019).

[25] Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. 2020. Towards robustness against natural language word substitutions. In *International Conference on Learning Representations*.

[26] Xin Dong, Yaxin Zhu, Yupeng Zhang, Zuohui Fu, Dongkuan Xu, Sen Yang, and Gerard De Melo. 2020. Leveraging adversarial training in self-learning for cross-lingual text classification. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1541–1544.

[27] Tianyu Du, Shouling Ji, Lujia Shen, Yao Zhang, Jinfeng Li, Jie Shi, Chengfang Fang, Jianwei Yin, Raheem Beyah, and Ting Wang. 2021. Cert-RNN: Towards Certifying the Robustness of Recurrent Neural Networks. (2021).

[28] Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On adversarial examples for character-level neural machine translation. *arXiv preprint arXiv:1806.09030* (2018).

[29] 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)*. Association for Computational Linguistics, Melbourne, Australia, 31–36. https://doi.org/10.18653/v1/P18-2006

[30] David Eppstein. 1995. Zonohedra and zonotopes. (1995).

[31] Chun Fan, Xiaoya Li, Yuxian Meng, Xiaofei Sun, Xiang Ao, Fei Wu, Jiwei Li, and Tianwei Zhang. 2021. Defending against Backdoor Attacks in Natural Language Generation. *arXiv preprint arXiv:2106.01810* (2021).

[32] Chaz Firestone. 2020. Performance vs. competence in human–machine comparisons. *Proceedings of the National Academy of Sciences* 117, 43 (2020), 26562–26571.

[33] 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*. IEEE Computer Society, 50–56. https://doi.org/10.1109/SPW.2018.00016

[34] Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970* (2020).

[35] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[36] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. arXiv:1412.6572 [stat.ML]

[37] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. 2018. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715* (2018).

[38] Junliang Guo, Zhirui Zhang, Linlin Zhang, Linli Xu, Boxing Chen, Enhong Chen, and Weihua Luo. 2021. Towards Variable-Length Textual Adversarial Attacks. *arXiv preprint arXiv:2104.08139* (2021).

[39] Wenjuan Han, Liwen Zhang, Yong Jiang, and Kewei Tu. 2020. Adversarial attack and defense of structured prediction models. *arXiv preprint arXiv:2010.01610* (2020).

[40] Xuanli He, Lingjuan Lyu, Qiongkai Xu, and Lichao Sun. 2021. Model Extraction and Adversarial Transferability, Your BERT is Vulnerable! *arXiv preprint arXiv:2103.10013* (2021).

[41] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In *NIPS*. 1693–1701.

[42] Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving Google's Perspective API Built for Detecting Toxic Comments. arXiv:1702.08138 [cs.LG]

[43] Yu-Lun Hsieh, Minhao Cheng, Da-Cheng Juan, Wei Wei, Wen-Lian Hsu, and Cho-Jui Hsieh. 2019. On the robustness of self-attentive models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 1520–1529.

[44] 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. *arXiv preprint arXiv:1909.01492* (2019).

[45] Aminul Huq, Mst Pervin, et al. 2020. Adversarial attacks and defense on texts: A survey. *arXiv preprint arXiv:2005.14108* (2020).

[46] Adam Ivankay, Ivan Girardi, Chiara Marchiori, and Pascal Frossard. 2022. Fooling Explanations in Text Classifiers. *arXiv preprint arXiv:2206.03178* (2022).

[47] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial Example Generation with Syntactically Controlled Paraphrase Networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, Marilyn A. Walker, Heng Ji, and Amanda Stent (Eds.). Association for Computational Linguistics, 1875–1885. https://doi.org/10.18653/v1/n18-1170

[48] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059* (2018).

[49] 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*. Association for Computational Linguistics, Copenhagen, Denmark, 2021–2031. https://doi.org/10.18653/v1/D17-1215

[50] Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328* (2017).

[51] Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. *arXiv preprint arXiv:1909.00986* (2019).

[52] 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* (2019).

[53] Erik Jones, Robin Jia, Aditi Raghunathan, and Percy Liang. 2020. Robust encodings: A framework for combating adversarial typos. *arXiv preprint arXiv:2005.01229* (2020).

[54] Aravind K. Joshi. 1985. Natural language parsing: Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions?

[55] Dan Jurafsky and James H. Martin. 2000. Speech and Language Processing.

[56] Dongyeop Kang, Tushar Khot, Ashish Sabharwal, and Eduard Hovy. 2018. Adventure: Adversarial training for textual entailment with knowledge-guided examples. *arXiv preprint arXiv:1805.04680* (2018).

[57] Sanjay Kariyappa and Moinuddin K Qureshi. 2019. Improving adversarial robustness of ensembles with diversity training. *arXiv preprint arXiv:1901.09981* (2019).

[58] Yannik Keller, Jan Mackensen, and Steffen Eger. 2021. BERT-Defense: A Probabilistic Model Based on BERT to Combat Cognitively Inspired Orthographic Adversarial Attacks. *arXiv preprint arXiv:2106.01452* (2021).

[59] James Kennedy and Russell Eberhart. 1995. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, Vol. 4. IEEE, 1942–1948.

[60] Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. 2022. Natural language processing: State of the art, current trends and challenges. *Multimedia Tools and Applications* (2022), 1–32.

[61] Ching-Yun Ko, Zhaoyang Lyu, Lily Weng, Luca Daniel, Ngai Wong, and Dahua Lin. 2019. POPQORN: Quantifying robustness of recurrent neural networks. In *International Conference on Machine Learning*. PMLR, 3468–3477.

[62] Zixiao Kong, Jingfeng Xue, Yong Wang, Lu Huang, Zequn Niu, and Feng Li. 2021. A Survey on Adversarial Attack in the Age of Artificial Intelligence. *Wireless Communications and Mobile Computing* 2021 (2021).

[63] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information* 10, 4 (2019), 150.

[64] Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and S. Ermon. 2018. Adversarial Examples for Natural Language Classification Problems.

[65] Emanuele La Malfa and Marta Kwiatkowska. 2022. The king is naked: on the notion of robustness for natural language processing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 11047–11057.

[66] Emanuele La Malfa, Min Wu, Luca Laurenti, Benjie Wang, Anthony Hartshorn, and Marta Kwiatkowska. 2020. Assessing robustness of text classification through maximal safe radius computation. *arXiv preprint arXiv:2010.02004* (2020).

[67] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. 2019. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 656–672.

[68] Hung-yi Lee, Shang-Wen Li, and Ngoc Thang Vu. 2022. Meta Learning for Natural Language Processing: A Survey. *arXiv preprint arXiv:2205.01500* (2022).

[69] Piyawat Lertvittayakumjorn and Francesca Toni. 2021. Explanation-based human debugging of nlp models: A survey. *Transactions of the Association for Computational Linguistics* 9 (2021), 1508–1528.

[70] Alexander Levine and Soheil Feizi. 2020. Robustness certificates for sparse adversarial attacks by randomized ablation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 4585–4593.

[71] Ang Li, Fangyuan Zhang, Shuangjiao Li, Tianhua Chen, Pan Su, and Hongtao Wang. 2023. Efficiently generating sentence-level textual adversarial examples with Seq2seq Stacked Auto-Encoder. *Expert Systems with Applications* 213 (2023), 119170.

[72] Jinfeng Li, Tianyu Du, Shouling Ji, Rong Zhang, Quan Lu, Min Yang, and Ting Wang. 2020. Textshield: Robust text classification based on multimodal embedding and neural machine translation. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*. 1381–1398.

[73] Jinfeng Li, Tianyu Du, Xiangyu Liu, Rong Zhang, Hui Xue, and Shouling Ji. 2021. Enhancing Model Robustness by Incorporating Adversarial Knowledge into Semantic Representation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7708–7712.

[74] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271* (2018).

[75] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[76] Linjie Li, Jie Lei, Zhe Gan, and Jingjing Liu. 2021. Adversarial VQA: A New Benchmark for Evaluating the Robustness of VQA Models. *arXiv preprint arXiv:2106.00245* (2021).

[77] Linyang Li and Xipeng Qiu. 2020. TAVAT: Token-Aware Virtual Adversarial Training for Language Understanding. *arXiv preprint arXiv:2004.14543* (2020).

[78] Lianjie Li, Zi Zhu, Dongyu Du, Shuxia Ren, Yao Zheng, and Guangsheng Chang. 2020. Adversarial Convolutional Neural Network for Text Classification. In *Proceedings of the 2020 4th International Conference on Electronic Information Technology and Computer Engineering*. 692–696.

[79] Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S Yu, and Lifang He. 2020. A survey on text classification: From shallow to deep learning. *arXiv preprint arXiv:2008.00364* (2020).

[80] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957* (2017).

[81] Zongyi Li, Jianhan Xu, Jiehang Zeng, Linyang Li, Xiaoqing Zheng, Qi Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. 2021. Searching for an Effective Defender: Benchmarking Defense against Adversarial Word Substitution. *arXiv preprint arXiv:2108.12777* (2021).

[82] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep Text Classification Can be Fooled. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, Jérôme Lang (Ed.). ijcai.org, 4208–4215. https://doi.org/10.24963/ijcai.2018/585

[83] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep Text Classification Can Be Fooled. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (Stockholm, Sweden) *(IJCAI'18)*. AAAI Press, 4208–4215.

[84] Hui Liu, Yongzheng Zhang, Yipeng Wang, Zheng Lin, and Yige Chen. 2020. Joint character-level word embedding and adversarial stability training to defend adversarial text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8384–8391.

[85] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. *arXiv preprint arXiv:1704.05742* (2017).

[86] Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994* (2020).

[87] Xingjun Ma, Yuhao Niu, Lin Gu, Yisen Wang, Yitian Zhao, James Bailey, and Feng Lu. 2021. Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition* 110 (2021), 107332.

[88] Pasquale Minervini and Sebastian Riedel. 2018. Adversarially regularising neural NLI models to integrate logical background knowledge. *arXiv preprint arXiv:1808.08609* (2018).

[89] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725* (2016).

[90] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence* 41, 8 (2018), 1979–1993.

[91] John X 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. *arXiv preprint arXiv:2005.05909* (2020).

[92] Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. *arXiv preprint arXiv:1603.00892* (2016).

[93] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 1797–1807. https://doi.org/10.18653/v1/D18-1206

[94] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial NLI: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599* (2019).

[95] Chidozie Nwafor et al. 2021. An Automated Mulitple-choice Question Generation Using Natural Language Processing Techniques. *International Journal on Natural Language Computing (IJNLC) Vol* 10 (2021).

[96] Daniel W Otter, Julian R Medina, and Jugal K Kalita. 2020. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems* 32, 2 (2020), 604–624.

[97] Mesut Ozdag. 2018. Adversarial attacks and defenses against deep neural networks: a survey. *Procedia Computer Science* 140 (2018), 152–161.

[98] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[99] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1532–1543. https://doi.org/10.3115/v1/d14-1162

[100] Lis Pereira, Xiaodong Liu, Hao Cheng, Hoifung Poon, Jianfeng Gao, and Ichiro Kobayashi. 2021. Targeted adversarial training for natural language understanding. *arXiv preprint arXiv:2104.05847* (2021).

[101] Yada Pruksachatkun, Satyapriya Krishna, Jwala Dhamala, Rahul Gupta, and Kai-Wei Chang. 2021. Does Robustness Improve Fairness? Approaching Fairness with Word Substitution Robustness Methods for Text Classification. *arXiv preprint arXiv:2106.10826* (2021).

[102] Danish Pruthi, Bhuwan Dhingra, and Zachary C Lipton. 2019. Combating adversarial misspellings with robust word recognition. *arXiv preprint arXiv:1905.11268* (2019).

[103] Shilin Qiu, Qihe Liu, Shijie Zhou, and Wen Huang. 2022. Adversarial attack and defense technologies in natural language processing: A survey. *Neurocomputing* 492 (2022), 278–307.

[104] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344* (2018).

[105] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* (2016).

[106] Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2021. Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages. arXiv:2104.05596 [cs.CL]

[107] Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The Curious Case of Hallucinations in Neural Machine Translation. In *NAACL-HLT*. Association for Computational Linguistics, 1172–1183.

[108] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. 2020. Adversarial attacks and defenses in deep learning. *Engineering* 6, 3 (2020), 346–360.

[109] 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*. 1085–1097.

[110] Yankun Ren, Jianbin Lin, Siliang Tang, Jun Zhou, Shuang Yang, Yuan Qi, and Xiang Ren. 2020. Generating natural language adversarial examples on a large scale with generative models. *arXiv preprint arXiv:2003.10388* (2020).

[111] Ishai Rosenberg, Asaf Shabtai, Yuval Elovici, and Lior Rokach. 2021. Sequence Squeezing: A Defense Method Against Adversarial Examples for API Call-Based RNN Variants. In *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–10.

[112] Allen Roush and Arvind Balaji. 2020. DebateSum: A large-scale argument mining and summarization dataset. In *Proceedings of the 7th Workshop on Argument Mining*. Association for Computational Linguistics, Online, 1–7. https://aclanthology.org/2020.argmining-1.1

[113] Cynthia Rudin and Joanna Radin. 2019. Why are we using black box models in AI when we don't need to? A lesson from an explainable AI competition. (2019).

[114] Wonryong Ryou, Jiayu Chen, Mislav Balunovic, Gagandeep Singh, Andrei Dan, and Martin Vechev. 2021. Scalable Polyhedral Verification of Recurrent Neural Networks. In *International Conference on Computer Aided Verification*. Springer, 225–248.

[115] Ananya B Sai, Akash Kumar Mohankumar, Siddhartha Arora, and Mitesh M Khapra. 2020. Improving dialog evaluation with a multi-reference adversarial dataset and large scale pretraining. *Transactions of the Association for Computational Linguistics* 8 (2020), 810–827.

[116] Keisuke Sakaguchi, Kevin Duh, Matt Post, and Benjamin Van Durme. 2017. Robsut wrod reocginiton via semi-character recurrent neural network. In *Thirty-first AAAI conference on artificial intelligence*.

[117] Elizabeth Salesky, Marcello Federico, and Marta Costa-jussà (Eds.). 2022. *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*. Association for Computational Linguistics, Dublin, Ireland (in-person and online). https://aclanthology.org/2022.iwslt-1.0

[118] Suranjana Samanta and Sameep Mehta. 2017. Towards Crafting Text Adversarial Samples. arXiv:1707.02812 [cs.LG]

[119] Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Interpretable adversarial perturbation in input embedding space for text. *arXiv preprint arXiv:1805.02917* (2018).

[120] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. 2019. Adversarial training for free! *Advances in Neural Information Processing Systems* 32 (2019).

[121] Zhouxing Shi, Huan Zhang, Kai-Wei Chang, Minlie Huang, and Cho-Jui Hsieh. 2020. Robustness verification for transformers. *arXiv preprint arXiv:2002.06622* (2020).

[122] Joe Stacey, Pasquale Minervini, Haim Dubossarsky, Sebastian Riedel, and Tim Rocktäschel. 2020. Avoiding the hypothesis-only bias in natural language inference via ensemble adversarial training. *arXiv preprint arXiv:2004.07790* (2020).

[123] Ieva Staliūnaitė, Philip John Gorinski, and Ignacio Iacobacci. 2021. Improving Commonsense Causal Reasoning by Adversarial Training and Data Augmentation. *arXiv preprint arXiv:2101.04966* (2021).

[124] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. 2017. Certified defenses for data poisoning attacks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 3520–3532.

[125] Bin Sun and Kan Li. 2021. Neural Dialogue Generation Methods in Open Domain: A Survey. *Natural Language Processing Research* 1, 3-4 (2021), 56–70.

[126] Peng Sun, Xuezhen Yang, Xiaobing Zhao, and Zhijuan Wang. 2018. An overview of named entity recognition. In *2018 International Conference on Asian Language Processing (IALP)*. IEEE, 273–278.

[127] Abigail Swenor and Jugal Kalita. 2022. Using Random Perturbations to Mitigate Adversarial Attacks on Sentiment Analysis Models. *arXiv preprint arXiv:2202.05758* (2022).

[128] Amirsina Torfi, Rouzbeh A Shirvani, Yaser Keneshloo, Nader Tavaf, and Edward A Fox. 2020. Natural language processing advancements by deep learning: A survey. *arXiv preprint arXiv:2003.01200* (2020).

[129] Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics* 7 (2019), 387–401.

[130] Eric Wallace, Mitchell Stern, and Dawn Song. 2020. Imitation attacks and defenses for black-box machine translation systems. *arXiv preprint arXiv:2004.15015* (2020).

[131] Matthew Wallace, Rishabh Khandelwal, and Brian Tang. 2022. Does IBP Scale? (2022).

[132] Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2020. Infobert: Improving robustness of language models from an information theoretic perspective. *arXiv preprint arXiv:2010.02329* (2020).

[133] Dilin Wang, Chengyue Gong, and Qiang Liu. 2019. Improving neural language modeling via adversarial training. In *International Conference on Machine Learning*. PMLR, 6555–6565.

[134] Tianlu Wang, Xuezhi Wang, Yao Qin, Ben Packer, Kang Li, Jilin Chen, Alex Beutel, and Ed Chi. 2020. CAT-Gen: Improving Robustness in NLP Models via Controlled Adversarial Text Generation. *arXiv preprint arXiv:2010.02338* (2020).

[135] Wenjie Wang, Pengfei Tang, Jian Lou, and Li Xiong. 2021. Certified Robustness to Word Substitution Attack with Differential Privacy. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1102–1112.

[136] Wenqi Wang, Run Wang, Jianpeng Ke, and Lina Wang. 2021. TextFirewall: Omni-Defending Against Adversarial Texts in Sentiment Classification. *IEEE Access* 9 (2021), 27467–27475.

[137] Wenqi Wang, Run Wang, Lina Wang, Zhibo Wang, and Aoshuang Ye. 2019. Towards a robust deep neural network in texts: A survey. *arXiv preprint arXiv:1902.07285* (2019).

[138] Wenqi Wang, Run Wang, Lina Wang, Zhibo Wang, and Aoshuang Ye. 2021. Towards a robust deep neural network against adversarial texts: A survey. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[139] Xiaosen Wang, Hao Jin, Yichen Yang, and Kun He. 2021. Natural Language Adversarial Defense through Synonym Encoding. (2021).

[140] Xiaosen Wang, Yifeng Xiong, and Kun He. 2021. Randomized Substitution and Vote for Textual Adversarial Example Detection. *arXiv preprint arXiv:2109.05698* (2021).

[141] Yicheng Wang and Mohit Bansal. 2018. Robust Machine Comprehension Models via Adversarial Training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 575–581. https://doi.org/10.18653/v1/N18-2091

[142] Yicheng Wang and Mohit Bansal. 2018. Robust machine comprehension models via adversarial training. *arXiv preprint arXiv:1804.06473* (2018).

[143] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. 2018. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv preprint arXiv:1801.10578* (2018).

[144] Adina Williams, Tristan Thrush, and Douwe Kiela. 2020. Anlizing the adversarial natural language inference dataset. *arXiv preprint arXiv:2010.12729* (2020).

[145] Jing Wu, Mingyi Zhou, Ce Zhu, Yipeng Liu, Mehrtash Harandi, and Li Li. 2021. Performance Evaluation of Adversarial Attacks: Discrepancies and Solutions. *arXiv preprint arXiv:2104.11103* (2021).

[146] Yi Wu, David Bamman, and Stuart Russell. 2017. Adversarial training for relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 1778–1783.

[147] Zhouhang Xie, Jonathan Brophy, Adam Noack, Wencong You, Kalyani Asthana, Carter Perkins, Sabrina Reis, Sameer Singh, and Daniel Lowd. 2022. Identifying Adversarial Attacks on Text Classifiers. *arXiv preprint arXiv:2201.08555* (2022).

[148] Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K Jain. 2020. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing* 17, 2 (2020), 151–178.

[149] Jingjing Xu, Liang Zhao, Hanqi Yan, Qi Zeng, Yun Liang, and Xu Sun. 2019. LexicalAT: Lexical-based adversarial reinforcement training for robust sentiment classification. 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)*. 5518–5527.

[150] Ying Xu, Xu Zhong, Antonio Jimeno Yepes, and Jey Han Lau. 2021. Grey-box Adversarial Attack And Defence For Sentiment Classification. *arXiv preprint arXiv:2103.11576* (2021).

[151] Ying Xu, Xu Zhong, Antonio Jose Jimeno Yepes, and Jey Han Lau. 2020. Elephant in the room: An evaluation framework for assessing adversarial examples in nlp. *arXiv preprint arXiv:2001.07820* (2020).

[152] Vikas Yadav and Steven Bethard. 2019. A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470* (2019).

[153] Shuoheng Yang, Yuxin Wang, and Xiaowen Chu. 2020. A survey of deep learning techniques for neural machine translation. *arXiv preprint arXiv:2002.07526* (2020).

[154] Yichen Yang, Xiaosen Wang, and Kun He. 2022. Robust Textual Embedding against Word-level Adversarial Attacks. *arXiv preprint arXiv:2202.13817* (2022).

[155] Ziqing Yang, Yiming Cui, Chenglei Si, Wanxiang Che, Ting Liu, Shijin Wang, and Guoping Hu. 2021. Adversarial Training for Machine Reading Comprehension with Virtual Embeddings. *arXiv preprint arXiv:2106.04437* (2021).

[156] Michihiro Yasunaga, Jungo Kasai, and Dragomir Radev. 2017. Robust multilingual part-of-speech tagging via adversarial training. *arXiv preprint arXiv:1711.04903* (2017).

[157] Mao Ye, Chengyue Gong, and Qiang Liu. 2020. SAFER: A structure-free approach for certified robustness to adversarial word substitutions. *arXiv preprint arXiv:2005.14424* (2020).

[158] KiYoon Yoo, Jangho Kim, Jiho Jang, and Nojun Kwak. 2022. Detection of Word Adversarial Examples in Text Classification: Benchmark and Baseline via Robust Density Estimation. *arXiv preprint arXiv:2203.01677* (2022).

[159] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems* 30, 9 (2019), 2805–2824.

[160] Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2019. Word-level textual adversarial attacking as combinatorial optimization. *arXiv preprint arXiv:1910.12196* (2019).

[161] Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Zixian Ma, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2020. Openattack: An open-source textual adversarial attack toolkit. *arXiv preprint arXiv:2009.09191* (2020).

[162] Jiehang Zeng, Xiaoqing Zheng, Jianhan Xu, Linyang Li, Liping Yuan, and Xuanjing Huang. 2021. Certified Robustness to Text Adversarial Attacks by Randomized [MASK]. *arXiv preprint arXiv:2105.03743* (2021).

[163] Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. 2020. Macer: Attack-free and scalable robust training via maximizing certified radius. *arXiv preprint arXiv:2001.02378* (2020).

[164] Chaoning Zhang, Philipp Benz, Chenguo Lin, Adil Karjauv, Jing Wu, and In So Kweon. 2021. A survey on universal adversarial attack. *arXiv preprint arXiv:2103.01498* (2021).

[165] Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2020. Generating fluent adversarial examples for natural languages. *arXiv preprint arXiv:2007.06174* (2020).

[166] Wei Zhang, Qian Chen, and Yunfang Chen. 2020. Deep Learning Based Robust Text Classification Method via Virtual Adversarial Training. *IEEE Access* 8 (2020), 61174–61182.

[167] 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 (2020), 1–41.

[168] Yuhao Zhang, Aws Albarghouthi, and Loris D'Antoni. 2021. Certified Robustness to Programmable Transformations in LSTMs. *arXiv preprint arXiv:2102.07818* (2021).

[169] Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130* (2019).

[170] Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating Natural Adversarial Examples. arXiv:1710.11342 [cs.LG]

[171] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. 2016. Improving the robustness of deep neural networks via stability training. In *Proceedings of the ieee conference on computer vision and pattern recognition*. 4480–4488.

[172] Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 2019. Learning to discriminate perturbations for blocking adversarial attacks in text classification. *arXiv preprint arXiv:1909.03084* (2019).

[173] Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-wei Chang, and Xuanjing Huang. 2020. Defense against adversarial attacks in nlp via dirichlet neighborhood ensemble. *arXiv preprint arXiv:2006.11627* (2020).

[174] Bin Zhu, Zhaoquan Gu, Le Wang, and Zhihong Tian. 2021. TREATED: Towards Universal Defense against Textual Adversarial Attacks. *arXiv preprint arXiv:2109.06176* (2021).

[175] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2019. Freelb: Enhanced adversarial training for natural language understanding. *arXiv preprint arXiv:1909.11764* (2019).

[176] Simiao Zuo, Chen Liang, Haoming Jiang, Xiaodong Liu, Pengcheng He, Jianfeng Gao, Weizhu Chen, and Tuo Zhao. 2021. Adversarial Training as Stackelberg Game: An Unrolled Optimization Approach. *arXiv preprint arXiv:2104.04886* (2021).