Robust Natural Language Processing: Recent Advances, Challenges, and Future Directions

Marwan Omar, Soohyeon Choi, DaeHun Nyang, and David Mohaisen

***Abstract*—Recent natural language processing (NLP) techniques have accomplished high performance on benchmark datasets, primarily due to the significant improvement in the performance of deep learning. The advances in the research community have led to great enhancements in state-of-the-art production systems for NLP tasks, such as virtual assistants, speech recognition, and sentiment analysis. However, such NLP systems still often fail when tested with adversarial attacks. The initial lack of robustness exposed troubling gaps in current models' language understanding capabilities, creating problems when NLP systems are deployed in real life. In this paper, we present a structured overview of NLP robustness research by summarizing the literature in a systemic way across various dimensions. We then take a deep-dive into the various dimensions of robustness, across techniques, metrics, embeddings, and benchmarks. Finally, we argue that robustness should be multi-dimensional, provide insights into current research, identify gaps in the literature to suggest directions worth pursuing to address these gaps.**

Index terms: Natural Language Processing; Adversarial Attacks; Robustness.

I. Introduction

Over the last decade, machines talking and interacting with humans in a human-like manner have become a reality. This reality is seen in many human-facing and emerging applications, including smart assistants, intelligent search engines, customer support, etc. Despite the significant differences among those applications in their associated contexts, they technically have one thing in common: they utilize an engine that employs advances in natural language processing (NLP) techniques, employing breakthroughs in deep machine learning (ML) and artificial intelligence (AI). As ML/AI continue to revolutionize our lives, leveraging and harnessing their power to understand natural languages, process and analyze them, and draw meaning from such analyses are the main promise that NLP applications aspire to deliver [1]. Technically, NLP today is a subarea of AI that allows machines read, understand, and obtain meanings from vast language artifacts.

NLP?s key benefits lie at the heart of teaching computers how to analyze large amounts of textual data. Although it may seem like a new technology, with the emergence of recent successful applications, NLP's roots go back to the early 1950s when NLP was first used for machine translation (MT) [2]. With the technology wave we are experiencing leading to innovation and disruptive applications, the amount of text data being generated everyday grows exponentially. This, in turn, created the need for powerful technologies, such as NLP, for efficiently processing voluminous amounts of data. NLP is being widely adopted by many industries to provide meaningful interpretation for data and help solving numerous challenges [3]. Moreover, NLP applications can be seen in our everyday life, e.g., Google Translate, Google Assistance, Amazon Alexa, Microsoft?s Cortana. In the financial industry, NLP is used in Prudential?s chat bot, Bank of America's Erica, among others. In the enterprise, NLP systems are widely adopted for the detection of spam, intrusions, malware, etc.

Machine learning techniques in general, and NLP techniques by association, are prone to attacks. Those attacks allow an adversary to target those techniques, as part of the aforementioned applications, to violate the application objectives and guarantees [4]. For example, in smart speakers applications, it has been shown that an adversary use minimally-modified inputs to trigger wrong, and sometimes malicious, device activation through voice squatting [5]–[8]. Similarly, an adversary might attack an NLP model that handles spam detection and fool it to make false predictions leading to spam passing through mail filters [9]. Malware authors might attack an NLP-based model to fool an intrusion detection system and missclassify malware as benign software [10]. Adversaries might even be more tempted to attack NLP models making decisions on loans in the finance industry with the incentive to fool a loan application system to incorrectly qualify a customer for a loan or vise-versa [11].

The research community has produced various studies demonstrating that NLP models are vulnerable to adversarial (machine learning) attacks, as NLP models are susceptible to making incorrect predictions on adversarial examples [12]. This, in turn, has led to a growing body of research on investigating and understanding the robustness of NLP techniques against adversarial attacks. Broadly speaking, such efforts in the literature are either focused on developing new attacks or better training models to make models resistant to such attacks (i.e., defenses) [13]. To sum up the research efforts dedicated understanding robustness in the literature, there are several research surveys that have addressed specific aspects of NLP robustness, e.g., data augmentation [14], search methods [15], pretrained models [16], and adversarial attacks [17]. However, the literature lacks research studies that provide a systematic overview of the state-of-the-art in this space across a range of variables; applications, technique, metrics, benchmark datasets, threat models, tasks, embedding techniques, learning techniques, goals, defense mechanisms, and performance.

Motivated by the lack of a pipeline-oriented view of the

M. Omar, S. Choi, and D. Mohaisen are with the Department of Computer Science at the University of Central Florida, Orlando, Florida 32816, USA. D. Nyang is with Ewha Womans University, Seoul, Republic of Korea. D. Mohaisen is the corresponding author (e-mail: mohaisen@ucf.edu; phone: +1-407-823-1294).

literature in the domain of NLP robustness, we sample and provide a systematic overview of the body of work done thus far in this space addressing this problem, and the novel aspect is a categorization that goes beyond what has been done in the literature. The main objective of this effort is to provide a road-map to the existing work, particularly over the past few years, and the research gap that deserves further attention through investigation. We note that most of the work on NLP robustness in the past three years tackled the issue from one angle and provides partial solutions rather than a unified and comprehensive framework on how to fix weaknesses [18]–[28]. Through this work, we wish to motivate the research community to develop a comprehensive frameworks to evaluate NLP robustness, in a pipeline (i.e., as envisioned to be deployed in a real application, tackling various use model aspects and building blocks). In particular, such a framework should enable analysis and probing to disclose NLP-models' strengths and weaknesses and provide recommendations on how to address weaknesses. Moreover, we envision that any proposed solution should provide us the ability to visualize, analyze, and extensively test NLP models for robustness by utilizing state-of-the art tools, against a range of settings.

Contribution. The main goal of this work is a fresh and deep look into the recent work on NLP robustness. To this end, this work makes the following contributions. (1) We introduce an enriched taxonomy that covers a range of dimensions of significant importance, driven from a pipeline of a broad range of NLP application. (2) We provide a categorization of various recent studies addressing NLP robustness, falling under the range of studied variables; e.g., models, embedding techniques, metrics, and techniques, among others. (3) We provide a contrast between the different approaches and their strengths and weaknesses. (4) We provide a road-map of the gaps left by the existing literature and call for actions.

Overall, this work offers researchers the ability to seek robustness from numerous aspects, e.g., choice of learning technique/model, embedding technique, datasets, defense mechanisms, and robustness metrics.

Organization. The rest of the paper is organized as follows. In section II, we review the related work in the context of NLP robustness (surveys). In section III, we present an overview of a generic pipeline to guide our review. In section IV, we provide a detailed review of the robustness techniques explored in the literature, which is the central theme of this work. In section V, we discuss the various metrics used for assessing robustness. In section VI, we highlight defence mechanisms for NLP. In section VII, we discuss the impact of embedding on robustness, while the impact of dataset is covered in section VIII. We conclude in section IX.

II. Related work

This paper is a survey in nature, and there has been several surveys addressing robustness of NLP techniques, as mentioned earlier. However, those surveys are narrow in scope, and address only a narrow aspect of the robustness spectrum. For the completeness of treatment of the subject, we address those related surveys in the following.

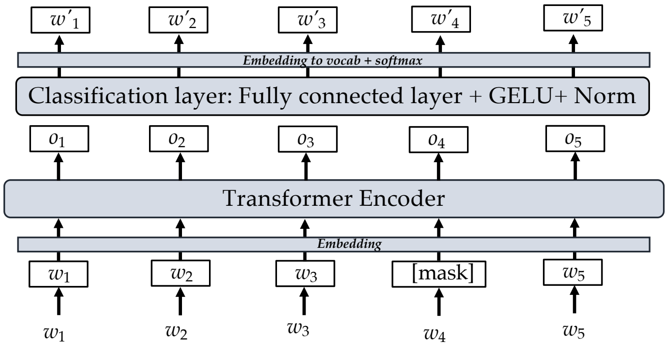

Fig. 1. A typical NLP pipeline, using the BERT embedding technique.

Feng *et al.* [14] conducted an extensive survey on data augmentation for NLP robustness. They studied various data augmentation techniques, including rule-based and model-based techniques as strategies to robustify NLP models to adversarial attacks. However, their work is limited in the sense that it only addresses a narrow aspect of robustness. Data augmentation is only part of several defense mechanisms to robustify NLP systems. For instance, there is robust data training and certifiable data training techniques, all of which are defense mechanisms to achieve robustness.

Yoo *et al.* [15] presented review of search algorithms for generating adversarial examples as a means to achieve robustness. However, their work is limited in scope in that it only focuses on adversarial examples as a means to seek robustness. There are many other variables in the robustness landscape such as embedding technique, robustness metrics, and robustness techniques which were not covered under their survey. Lin *et al.* [29] presented a research survey on transformers which are pre-trained NLP models. The study is centered around robustness from a model stand point where learning models such as BERT and RoBERTa can contribute to robustness. However, their work falls short in addressing other aspects, e.g., defenses, attacks, and techniques.

In [17], Zhang *et al.* conducted a survey on adversarial attacks as a means to evaluate robustness of NLP systems to adversarial perturbations. However, their work is limited due to the fact that robustness is multi-dimensional and adversarial attacks are only one dimension of robustness. Robustness entails numerous other elements such as defenses, metrics, and embedding technique. To the best of our knowledge, there is no comprehensive work in the literature that puts together advances on understanding the robustness considering a pipeline that accounts for the important steps in implementing an NLP system, which is our take in this work.

III. NLP: A generic overview

To simplify our presentation of the overview of the various advances made over the past decade in the area of NLP towards improving our understanding of robustness through attacks and defenses, we highlight a system flow with various elements that are typical in NLP systems. We use those elements to describe the different advances. In the subsequent

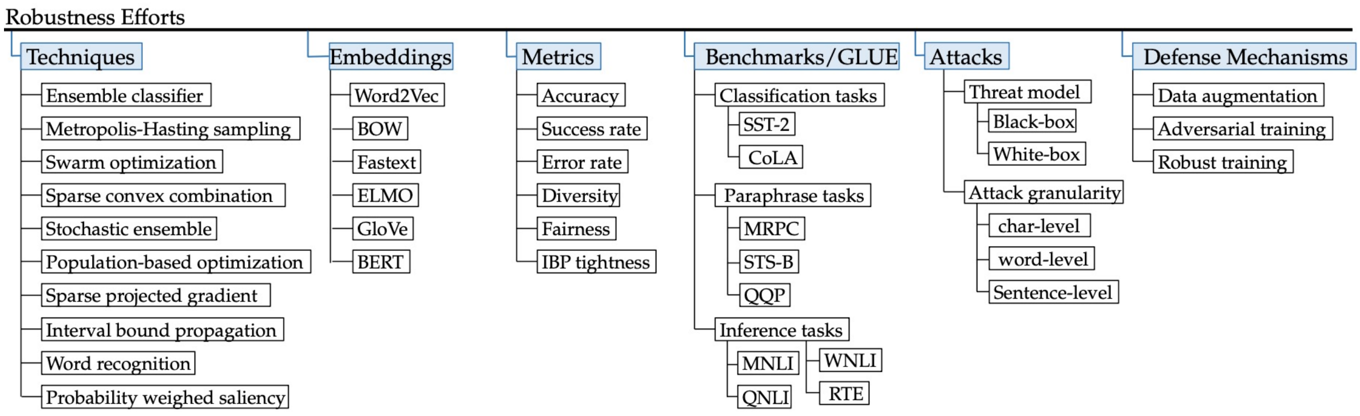

Fig. 2. A high-level overview of the various research efforts in the domain of robustness analysis across various elements of the NLP pipeline, including techniques, embedding, metrics, benchmarks, attack model, and defense mechanisms.

description, we envision the application of a natural language model (used for natural language generation).

At a high-level, and is exemplified later in Figure 1, the typical NLP pipelines consist of a preprocessing step that takes a raw language input and prepares it for consumption by an NLP model through the appropriate steps of mapping. Upon that initial step of preprocessing and mapping, an embedding step is invoked to transform the initial representation into an appropriate format that can be consumed by the model. The model then runs, initially in a training phase, to learn several parameters that could be used for the language generation.

NLP Robustness: Taxonomy. In Figure 2, we demonstrate a brief taxonomy of the various efforts presented in the literature on NLP and associated robustness analysis across the associated pipeline, including techniques, embedding, evaluation metrics, evaluation benchmarks (datasets), attack space (threat model and granularity), and associated defense mechanisms.

Given the range of objectives that each paper tries to address, there is a need to systematically understand them by breaking them down into some normal form, based on the pipeline we described here. As such, in the following sections, we dive into efforts that have been dedicated on each among those elements of the pipeline.

To make this survey more accessible, we compile the acronyms used in the rest of this paper in Table I.

IV. Robustness Techniques and Tools

A. Background

As we pointed out in section I, NLP models are being increasingly deployed to handle many human-like tasks. With the prevalence of NLP systems and their various real-world applications, the need for building robust NLP models becomes paramount because the consequences of making false predictions can be detrimental and even life-threatening in some cases (e.g., medical imaging diagnosis systems). In reality, however, we have seen numerous examples of failed NLP models after deployment in the real-world due to the lack of robustness. Some recent work has shown that approximately two-thirds of real-world NLP systems (e.g., essay grading NLP, Microsoft twitter NLP) fail after deployment due to the lack of robustness [18]. As a case in point, Amazon built an NLP-based recruiting tool that was deemed as a failure (and was eventually scraped) because the NLP-powered model demonstrated bias against female applicants [30].

To address these issues, the security and NLP communities alike invested a significant amount of effort in developing techniques and tools for testing and analyzing the robustness of language processing techniques, embedding techniques that would provide better robustness, metrics to assess the performance, benchmarks to evaluate this robustness, attacks to challenge the robustness, and the associated defense mechanisms. In the following, we sample and review efforts in each of those directions, using the reference taxonomy in Figure 2.

B. Robustness Analysis and Testing Tools

As a result of the failure of various NLP systems, the research community has conducted numerous studies on NLP robustness, calling for extensive testing of NLP models before deployment. For example, Rebeiro *et al.* [18] introduced CheckList, a task-agnostic method for studying NLP models. CheckList incorporates a matrix of general linguistic capabilities and test types that allow for comprehensive test iteration. The proposed approach works on both commercial as well as research NLP models, and reveals model weaknesses even after models' internal testing, although stops short of providing solutions for the identified weaknesses.

Similarly, Goel *et al.* [25] identified challenges in evaluating NLP systems. As a result, they introduced a solution called Robustness Gym (RG), which is a simple yet extensible evaluation toolkit. By realizing a common mean for evaluation, RG enables NLP practitioners to compare various results from various frameworks and to develop new methods using a built-in sets of abstractions. Moreover, RG offers a unified NLP-model evaluation framework that allows thorough and extensive analysis and test of NLP models. While promising, the robustness framework does not seem to offer insights into the full understanding of the behavior of NLP models nor disclosing where the systems are actually failing or performing well. It would seem both desired and perhaps intuitive to extend the tool to localize the model failure and provide reasons behind model degradation.

In [24], Rychalska *et al.* introduced WildNLP, a framework for testing model stability in-situ. In WildNLP, text

TABLE I
LIST OF ACRONYMS IN ALPHABETICAL ORDER.

Term	Definition
AGNEWS	AG's News Topic Classification Dataset
AI	Artificial Intelligence
ASCC	Adversarial Sparse Convex Combination
BERT	Bidirectional Encoder Representations from Transformers
BoW	Bag-of-Words
CNN	Convolutional Neural Networks
CoLA	Corpus of Linguistic Acceptability
DNE	Dirichlet Neighborhood Ensemble
ELMO	Embeddings from Language Model
GELU	Gaussian Error Linear Unit
GloVe	Global Vectors for Word Representation
GLUE	Global Language Understanding Evaluation
IB	Information Bottleneck
IBP	Interval Bound Propagation
IMDB	Internet Movie Database
LSTM	Long Short Term Memory
MCMC	Markov Chain Monte Carlo
MH	Metropolis-Hastings
MHA	Metropolis-Hastings Sampling Algorithm
MHS	Metropolis-Hastings Sampling
ML	Machine Learning
MNLI	Multi-Genre Natural Language Inference
MRPC	Microsoft Research Paraphrase Corpus
MT	Machine Translation
MTL	Multi-Task Learning
NER	Named Entity Recognition
NLI	Natural Language Inference
NLP	Natural Language Processing
OOD	Out-Of-Distribution
PI	Paraphrase Identification
PSO	Particle Swarm Optimization Algorithm
PWWS	Probability Weighted Word Saliency
QA	Question-Answering
QNLI	Question-Answering Natural Language Inference
QQP	Quora Question Pairs
RG	Robustness Gym
RoBERTa	Robustly Optimized BERT Pretraining Approach
RTE	Recognizing Textual Entailment
SA	Sentiment Analysis
SEA	Semantically Equivalent Adversaries
Seq2Seq	Sequence-to-Sequence
SQuAD	Stanford Question Answering Dataset
SPGD	Sparse Projected Gradient Descent
SST-2	Stanford Sentiment Treebank – version 2
STS-B	Semantic Textual Similarity Benchmark
WNLI	Winograd Natural Language Inference

corruptions, such as keyboard errors or misspelling occur, are addressed. To this end, the authors compare the robustness of models in four popular NLP tasks: QA, NLI, NER, and SA. The authors do so by testing the performance of these tasks on aspects introduced in the framework, and find that the high performance of models does not guarantee sufficient robustness, although recent embedding techniques can help improve that. In order for us to improve the models robustness, we need to incorporate several factors rather than just simply using the adversarial attacks as a metric; e.g., the underlying model properties, test data, appropriate metrics, etc.

Datasets and their role in highlighting the performance of various algorithms, as well as unveiling their robustness, have been also examined. In [31], Hendrycks *et al.* systematically examined and measured the out-of-distribution (OOD) generalization for seven NLP datasets. En route, they construct a robustness benchmark that employs realistic distribution shifts and measure the generalization of various models, including Bag-of-Words (BoW) models, CNNs, and LSTMs. Moreover, they show that the performance of pretrained transformers? decline is substantially smaller. The authors also examined the factors that affect the robustness, and found that larger models are not necessarily more robust than smaller models, while more diverse pretraining data could improve the robustness. The authors also use the generalization benchmark to train a model on the SST-2 [32] dataset and evaluate on the IMDB [33] (both of which are popular benchmark datasets for the sentiment analysis task). They use BoW, CNN, and LSTM-based models to predict a movie review?s binary sentiment, and report the accuracy.

C. Techniques for Robustifying NLP Models

A number of techniques have been proposed in the literature for robustifying NLP models, including ensemble classifiers with randomized smoothing, stochastic ensembles, interval bound propagation, word recognition techniques, etc. In the following, we review some of the most widely used robustness techniques and where they are used. A summary of some of those works is shown in Table II. Additionally, the reader is referred to Figure 2 for additional context.

1) Ensemble Classifiers with Randomized Smoothing: One of the obvious caveats of relying on a single classifier in NLP tasks is that a manipulation of the underlying input space (feature) fed into this classifier would have a significant impact on the output of the classifier. To cope with this issue, ensemble classifiers are proposed in the machine learning literature, where multiple classifiers (estimators) are built independently and aggregated to obtain the final result of the classifier. As such, and without losing generality, an ensemble classifier is a classifier whose decision depends on the combined outcome of decisions made by several individual classifiers, and is a method for achieving a degree of robustness in NLP models by reducing bias in the training data.

By the same token, robustness with the ensemble classifier means that for any input x and class label y, a smoothed classifier (g) will return a prediction $g(x)$ which is most likely the correct prediction [63]. As such, a model is said to be robust at y if it can classify all inputs in the perturbation text correctly [64]. Randomized smoothing, on the other hand, is a method through which we can transform a classifier into a new smoothed classifier that is robust in a given setting. Randomized smoothing can provably certify the robustness of NLP models against various adversarial attacks such as word-substitution attacks [22]. For instance, Zhout *et al.* proposed Dirichlet Neighborhood Ensemble (DNE), a randomized smoothing method for training a robust model in a way that mitigates substitution-based attacks [60]. Essentially, DNE forms virtual sentences by sampling embedding vectors for each word in an input sentence from using a group of the word and its synonyms, and augments them with the training data. This sampling, in turn, ensures robustness against adversarial attacks without sacrificing the performance. While the randomized smoothing technique greatly enhanced classification accuracy against adversarial attacks, this technique only applies to one task at a time. In other words, if the technique

TABLE II

A COMPARISON BETWEEN VARIOUS WORKS FROM THE LITERATURE, ACROSS TECHNIQUES, WHETHER A BENCHMARK IS UTILIZED OR NOT, THE THREAT MODEL (WHITE-BOX VS BLACK-BOX), AND STUDY'S GOAL (ATTACK, DEFENSE OR ROBUSTNESS).

Study	Year	Technique	Benchmark	White-Box	Black-Box	Attack	Defense	Robustness
Carllini *et al.* [34]	2017	Defensive Distillation	✓	✓	✗	✗	✗	✓
Ebrahimi *et al.* [35]	2017	Character Substitution	✓	✓	✗	✓	✓	✓
Madry *et al.* [36]	2017	Projected Gradient Descent	✓	✓	✗	✗	✗	✓
Wong *et al.* [37]	2017	Text Example Generation	✓	✗	✓	✓	✗	✗
Zhao *et al.* [38]	2017	Stochastic Search Algorithm	✓	✗	✓	✓	✗	✗
Alzantot *et al.* [39]	2018	Population-based Optimization	✓	✗	✓	✓	✗	✗
Dong *et al.* [21]	2018	Sparse Convex Combination	✗	✗	✓	✗	✗	✓
Geiger *et al.* [40]	2018	Multiply-Quantified Sentences	✗	✓	✗	✓	✓	✗
Glochner *et al.* [41]	2018	Generating an NLI Test Set	✓	✓	✓	✓	✗	✗
Gong *et al.* [42]	2018	Nearest-neighbour Search	✓	✓	✗	✓	✓	✗
Braham *et al.* [43]	2019	Sparse Projected Gradient Decent	✗	✓	✗	✗	✗	✓
Clark *et al.* [44]	2019	Learned-mixin Ensemble	✗	✓	✗	✗	✓	✗
Eger *et al.* [45]	2019	Character Substitution	✗	✓	✗	✓	✗	✗
He *et al.* [46]	2019	Debiasing Data	✓	✓	✓	✓	✓	✗
Jia *et al.* [47]	2019	Interval Bound Propagation	✓	✓	✗	✗	✗	✓
Jung *et al.* [48]	2019	Biases in Summerization Analysis	✓	✓	✓	✗	✗	✓
Kaushik *et al.* [49]	2019	Spurious Correlation	✓	✓	✗	✓	✓	✗
Pruthi *et al.* [50]	2019	Word Recognition	✓	✓	✗	✗	✓	✗
Ren *et al.* [51]	2019	Probability Weighted Word Saliency	✓	✓	✗	✓	✗	✗
Yaghoobzadeh *et al.* [52]	2019	Forgettable Examples for Robustness	✗	✗	✓	✗	✗	✓
Zang *et al.* [53]	2019	Swarm Optimization Algorithm	✓	✓	✗	✓	✗	✗
Cheng *et al.* [54]	2020	Seq2Seq	✗	✓	✗	✓	✗	✗
Hendrycks *et al.* [31]	2020	Out of Distribution (OOD)	✓	✓	✗	✗	✗	✓
Jin *et al.* [55]	2020	Text Generation	✓	✗	✓	✓	✗	✗
Wang *et al.* [56]	2020	Information Bottleneck (IB) Regulizer	✗	✓	✗	✗	✗	✓
Wang *et al.* [57]	2020	Supervised Text Classification	✓	✓	✗	✓	✓	✗
Wang *et al.* [58]	2020	Controlled Generation	✗	✓	✗	✓	✓	✗
Ye *et al.* [20]	2020	Stochastic Ensemble	✓	✗	✓	✗	✓	✗
Zhang *et al.* [59]	2020	Metropolis-Hastings Sampling	✓	✓	✓	✓	✓	✗
Zhout *et al.* [60]	2020	Ensemble Classifier	✓	✓	✗	✗	✓	✗
Fabbri *et al.* [30]	2021	Text Summerization	✓	✓	✗	✓	✓	✗
Goel *et al.* [25]	2021	Heuristics	✗	✓	✗	✗	✗	✓
Schiller *et al.* [61]	2021	Stance Detection Benchmark	✗	✓	✗	✓	✓	✓
Xu *et al.* [62]	2021	Text Generation	✓	✓	✓	✓	✓	✓
Zeng *et al.* [19]	2021	RanMask	✓	✗	✓	✗	✗	✓

was to be applied to another task, e.g., NLI, the robust training procedure would need to be restarted from scratch. This, in turn, will incur a significant amount of overhead [64].

2) Stochastic Ensemble with Randomized Smoothing: Stochastic ensemble refers to a classifier with some randomness and uncertainty in the underlying model. NLP models, in general, have a stochastic characteristic which (if understood correctly) enables us to effectively characterize the behavior of the NLP predictive models [19]. In [20], Ye *et al.* proposed a certified robustness method based on a new randomized smoothing technique that constructs a stochastic ensemble by applying random word substitutions on the input sentences. Moreover, their method leverages the statistical properties of the ensemble to provably certify robustness.

This method is simple and generalizable in the sense that it does not depend on any structure and only requires black-box queries of the model outputs. As such, their method can be applied to any pre-trained model (e.g., BERT) and granularity (e.g., word-level, subword-level). Although *robust training* has been proven to enhance the overall robustness of the model against adversarial word-level perturbations, robust training on a different model requires re-executing the training steps from scratch which is one of the key limitations of certifiably robust training, and is an issue that this study fails to reveal.

3) Interval Bound Propagation: The Interval Bound Propagation (IBP) is a technique used to build certifiably robust machine learning classifiers. IBP essentially uses the interval arithmetic to define a loss to minimize an upper bound on the maximal difference between any pair of logits when the input is perturbed within any norm-bounded ball [65]. IBP has been applied widely and successfully in the vision domain to obtain robustness guarantees [65]–[69].

The key strength of IBP for NLP models is that it can be used to process discrete perturbations in addition to the continuous ones, which are used in the computer vision domain [64]. It is our belief the robustness cannot be understood in isolation of models obtained from different datasets, irrespective of the metric. As such, one obvious limitation of this line of work is that it did not use a broad benchmark; e.g., GLUE, which includes six benchmark datasets. Those datasets could have been possibly used to further evaluate the classification accuracy and demonstrate their robustness technique.

4) Word Recognition: Word recognition refers to an elementary process of language, whereby written and verbal forms of words are converted into linguistic tasks and representations. Word recognition as a technique has been used in various studies [50], [70]–[74]. For example, Pruthi *et al.* proposed using a word recognition model in front of the classifier to combat adversarial spelling mistakes against a BERT model used for sentiment analysis. They show that a single adversarially-chosen character attack has lowered the accuracy from 90.3% to 45.8%, while their defense brings the accuracy back up to 75%.

This approach can be easily used to robustify NLP models against adversarial spelling mistakes. Moreover, the same approach can be used for recognizing words corrupted by random keyboard mistakes, thus defending NLP models against word perturbation attacks. What is unique about this approach is that, unlike many other studies which only study adversarial attacks on NLP models, it demonstrates vulnerable state-of-the-art NLP model and also proposes a robustness technique that contributes to protection against the same attacks. A study that combines both attack and defense strategies is certainly plausible. On the other hand, a limitation of this approach is that it is unclear if it transfers and generalizes to other network architectures across various linguistic tasks.

D. Breaching Security by Improving Attacks

Adversarial attacks have been used in the literature to evaluate robustness of NLP systems to real-world attacks. As depicted in Figure 2, most research works tackle the adversarial attack issue from either an attack-granularity angle (character-level, word-level, and sentence-level attacks) or from a threat model (white-box and black-box) angle.

Numerous research studies have extensively studied the role of adversarial attacks in developing robust NLP models [35], [39], [54], [58]. For example, Cheng *et al.* [54] study crafting AEs for seq2seq models whose inputs are discrete text strings. To address the challenges caused by the discrete input space, the authors propose a projected gradient method combined with group lasso and gradient regularization in the white-box threat model. To handle the large output space, they design a new loss functions that works for deriving both non-overlapping and targeted keyword attacks. The authors achieve an average of 85% success rate with their adversarial attacks on NLP models, however, they do not indicate what attributed to the success of the attack: whether it is the poorly designed seq2seq model or is it the dataset. Also, authors stop short in offering any recommendations on how to increase the robustness of seq2seq models against adversarial attacks.

From an attack granularity perspective, various studies have been carried out using either a character-level, word-level, or sentence-level attack [37], [38], [45], [53], [58], [59], [75]. For example, Eger *et al.* [45] investigate the impact of visual adversarial attacks (modification to text which can be detected by visualizing) on character-, word-, and sentence-level tasks. They show that both neural and non-neural models, in contrast to humans, are vulnerable to such attacks, leading to a performance decrease of up to 82%. In the following subsection, we explore, in details, some of the attack methods/techniques used in the NLP robustness literature:

1) Adversarial Sparse Convex Combination (ASCC): Sparse convex combination refers to the method of representing the target output as a sparse convex combination of the input text. Based on this definition, for any input x and class label y, a trained NLP classification model maps each input x to its class label y. Given a clean (unperturbed) input x a targeted sparse adversarial attack aims for finding a perturbation so that the perturbed input x' is incorrectly classified to a target class [21]. This attack method has been used in numerous research studies [21], [70], [76]–[80].

For example, in [21], Dong *et al.* introduced an Adversarial Sparse Convex Combination (ASCC) method to model the word substitution attack space and leverage a regularization term to enforce perturbation towards an actual substitution. In doing so, they align their modeling better with the discrete textual space. Based on the ASCC method, they also generate worst-case perturbations and incorporate adversarial training for robustness. Their experiments show that ASCC-defense outperforms the current state-of-the-art techniques in terms of robustness on two prevailing NLP tasks, SA and NLI, and address several attacks across several architectures.

The strength of this attack method is that it can be used to generate adversarial examples to robustify models and eventually improve models' prediction accuracy. On the other hand, when robust accuracy on adversarial examples goes up, this causes the clean accuracy (unperturbed standard test data) to go down, a trade-off that should be considered when utilizing this attack method [64].

2) Population-based Optimization for Adversarial Attacks: A population-based optimization algorithm is a type of genetic algorithm that aims to find perturbations which can change a model?s prediction/classification [81], [82]. This technique maintains a "population" of candidates' inputs and continuously perturbs and combines them [64]. On the other hand, a black-box attack is a type of adversarial attack where an adversary does not have access to the model's internal structure nor parameters. This attack technique has been used in numerous research works [12], [39], [53], [83]–[89].

Alzantot *et al.* [39] proposed a population-based optimization algorithm to generate semantically and syntactically similar AEs that can fool SA and textual entailment models with high accuracy. Moreover, they demonstrate that more than 90% of the successful SA AEs are classified to their original label by 20 human annotators, and that the examples are perceptibly quite similar.

While this attack technique concretely examines the vulnerability of NLP models to adversarial examples, it falls short of proving whether the uncovered attack remains effective under various model architectures (transferability). For instance, it is unknown if the attack success rate would remain the same under the LSTM and Word-CNN models.

3) Sparse Projected Gradient Descent: The projected gradient descent method is a type of greedy algorithm which has been applied broadly to machine learning models [36]. In this method, each element in the input text is considered for substitution and the best perturbations are selected from all possible perturbations and rerun until no more perturbations are possible [15]. This attack method has been utilized in several research works [15], [43], [90] with various promising results. For example, Barham *et al.* [43] introduced a sparse projected gradient descent (SPGD) method for crafting interpretable AEs for text applications. SPGD imposes a regularization constraint on input perturbations by projecting them onto the directions to nearby word embeddings with the highest similarity.

The strength of this attack method is that the regularization constraint ensures that perturbations move each word embedding in an interpretable direction (i.e., towards another nearby word embedding) while ensuring a high prediction accuracy on different model architectures. A limitation of this attack method is that it is implemented using only one dataset, which is the IMDB. Ideally, robust NLP models should be evaluated across various linguistic tasks on multiple datasets. Research has shown that models should be tested using the GLUE (Global Language Understanding Evaluation) benchmark which includes nine datasets, six of which are for testing classification accuracy [90].

4) Probability Weighted Word Saliency (PWWS): PWWS is a greedy search method for generating adversarial examples [91]. In this method, the goal is to rank words based on some importance function. In a descending importance order, each word is replaced with a candidate word until we successfully perturb all words [15]. This technique has been used in multiple studies addressing machine learning robustness to adversarial attacks, including the studies in [51], [92]–[95]. The way such a technique is used in those studies is almost identical. For example, Ren *et al.* [51] addressed the problem of AEs on text classification by generating AEs that maintain lexical and grammatical correctness, as well as semantic similarity. Based on the synonyms substitution strategy, they introduce a word replacement order determined by both the word saliency and the classification probability, and propose a greedy PWWS algorithm for text AEs. Their experiments on three datasets using convolutional and LSTM based models show that their approach reduces the classification accuracy and keeps a very low word substitution rate. The strength of this technique is that it exposes the vulnerabilities of NLP models to adversarial examples via word replacement using an efficient greedy algorithm. One limitation of this technique, however, is that it does not consider the error analysis aspect of robustness. In other words, the technique does not indicate which models were correct on the original words/data but incorrect on the perturbed words.

5) Swarm Optimization Algorithm for Adversarial Attacks: The Particle Swarm Optimization Algorithm is a search algorithm used to generate adversarial examples [53]. In this method, each member of the population is perturbed by creating all potential candidate obtained by replacing each input and then sampling one input example, at each iteration. Using this algorithm, we are able to find the best perturbed input among all members of the population [15]. This attack technique has been used in multiple studies addressing machine learning robustness to adversarial attacks in general, including the studies in [12], [15], [53], [96], [97]. The way such a technique is used in those studies is almost identical; e.g., Zang *et al.* [53] propose an attack model that incorporates a word substitution method [98] and particle swarm optimization-based search algorithm for that purpose with a significant success.

The strength of this method is its generalization to different model architectures, such as BiLSTM and BERT, using benchmark datasets. Moreover, this attack method achieves higher attack success rates and crafts more high-quality AEs in comparison to various baseline methods.

A key limitation of this attack method, however, is that it does not take into account run-time of the PSO algorithm, as the run-time is a critical factor for search algorithms in any real-world deployment. Per [15], a key factor to consider for those algorithms' complexity is the length of the input text, as well as the choice of the search algorithm. For instance, if the input texts are short (e.g., a few sentences), a beam search is an appropriate choice, since it can achieve a high success rate without incurring the overhead. In such tasks, AEs must be generated quickly, and a more efficient algorithm may be preferred, even with a lower success rate.

6) Metropolis-Hastings Sampling for Adversarial Attacks: The Metropolis Hastings (MH) Sampling is a Markov Chain Monte Carlo (MCMC) algorithm for generating a sequence of random samples from a probability distribution where direct sampling is hard [59]. MH works by conducting a random walk according to a Markov chain whose stationary distribution is π (the eventual distribution from which the chain will sample). On each step of the MC, a new state is proposed and either accepted or rejected according to a dynamically calculated probability value, called the acceptance criteria [78]. That is, in the long term, the data points from the MC will look similar to the data points from π [60], [59].

This attack technique has been used in multiple studies addressing machine learning robustness to adversarial attacks, including [59], [99]–[101]. The way such a technique is used in those studies is almost identical. For example, Zhang *et al.* propose Metropolis-Hastings Sampling (MHS) to generate fluent adversarial examples for attacking NLP models. The authors perform Metropolis-Hastings sampling which is designed with the guidance of gradients. The strength of this method is that it outperforms the baseline models on attacking capability. However, this method was not tested under various model architectures as well as linguistic tasks. Ideally, an attack technique should scale up to re-use across numerous tasks (e.g., sentiment analysis and NLI) and with various model architectures (e.g., RoBERTa, LSTM, Attention-based, etc.).

E. Insights and Open Directions

It is evident that the attack methods utilized in the literature for evaluating the performance and robustness of NLP models are diverse. This diversity of attack methods is dictated by the variety of limitations that some of those attacks have, precluding their use in broad set of applications while advocating others for their strengths and ease of implementation. In general, NLP models have to be evaluated to determine their robustness to adversarial attacks, where different attacks may prove more potent than others.

Overall, our exploration of the attack methods space calls work in various directions to fill various gaps. (1) While there is a significant initial work on the utilization and implementation of various attack techniques in the broad NLP community, the attacks developed so far are limited in many ways, and there is a need for developing techniques that are transferable and can be generalized to various NLP model architectures. (2) Generally, and with a few exceptions, the majority of the work in the literature considers specific task definitions for which the robustness is analyzed and understood with respect to the proposed attacks. For instance, many of such efforts have focused on only a specific task, such as sentiment analysis, question answering, etc. and left unaddressed the challenge of defending NLP models against a generic adversary optimizing in the input language for multiple tasks [40], [41], [48], [49], [102]. Moreover, most research works focus on how to develop certain types of concrete adversarial examples in a constrained adversarial setting. This creates an open challenge, that is hopefully within reach, for the research community where attack techniques should consider an advanced adversary and take into account how a determined and unconstrained adversary might circumvent robustness measures put for NLP tasks. (3) State-of-the-art NLP systems are demonstrably vulnerable to adversarial attacks which, in turn, causes inaccuracies in their prediction capabilities. Although researchers have provably shown the vulnerable state of NLP models, they fall short, however, in identifying the gaps in models' capabilities and how to improve such models. It remains an open challenge for the NLP research community to conduct extensive experiments to deep-dive into the internal structure of NLP models to distinguish models which fare better or worse than others under adversarial attacks. For instance, it would be interesting to know if a BERT model fares better or worse than an LSTM (Long Short Term Memory) model. (4) Most of the research progress on NPL robustness techniques concerning the development of new attacks takes into consideration worst-case scenarios/examples without indicating which type of worst-case scenario to focus on. While that is understanding from a security standpoint, the comprehensive best-case and average-case analysis would be worthwhile, although unclear how tractable such an analysis would be.

V. Robustness Metrics

The robustness of NLP models is a quality that has to be measured with well-defined and relevant metrics in order to gain an understanding of the level of NLP model's resistance against adversarial attacks. In general, metrics serve a dual purpose in machine learning systems design: measuring their performance in training and testing. Robustness metrics are similar, in the sense that they are used for measuring and tracking the performance of the machine learning models under adversarial settings. Based on the surveyed literature, we found that different research works utilize different metrics for measuring robustness. A summary of some of those works is shown in Table II, with the context highlighted in Figure 2. In the following, we review some of those metrics.

A. Attack Success Rate

Attack success is one of the simplest and most widely utilized metrics for evaluating the robustness of NLP models. The attack success rate refers to the number of attempts that are successfully normalized by the number of overall attempts of an attack (e.g., number of valid adversarial examples that both meet a predefined example condition on the size of perturbation and the adversary's objective; e.g., reducing the confidence of a classifier below a given threshold, or changing the classification label of the example).

As a metric, the attack success rate has been utilized in numerous research studies to determine the effectiveness of adversarial attacks on NLP models [12], [15], [39], [115], [116]. For instance, Alzantot *et al.* [39] measured the effectiveness of their genetic algorithm-based adversarial attacks using the attack success rate as a metric which eventually indicates NLP model robustness to adversarial examples.

While the attack success rate is simple and easy to interpret metric, its main disadvantage is that it ignores most, if not all, quality characteristics of the resulting adversarial examples that contributed to the success rate. For instance, as we will see later, not all adversarial examples are considered of the same quality, where some of them may be easily eliminated or detected using simple heuristics while others are more challenging to address using the same heuristics.

B. Error Rate

Error rate (also known as the robustness error) refers to the number of times where an NLP model incorrectly classifies an input text. The error rate is a metric which has been used in numerous research studies to determine the robustness of NLP models to adversarial attacks [117]. In contrast to the attack success rate, the lower the error rate (misclassification rate), the more robust the NLP model is against adversarial attacks.

The error rate has been used as metric in numerous research studies, including [13], [37], [103]–[105]. For example, Goodfellow *et al.* [13] found that several models, including neural network-based models, consistently misclassify AEs inputs formed by applying small but intentionally worst-case perturbations to the input examples from a dataset. In doing so, the perturbed input forces the model to output an incorrect answer with high confidence. According to the authors, adversarial examples are often misclassified by a variety of classifiers with different architectures.

This metric is simple and easy to calculate in order to evaluate the robustness of NLP models to adversarial attacks. However, the error rate alone should not be the only metric to evaluate the performance of machine learning models as it does not take into account the intrinsic and often clear differences between the examples contributing to the error rate.

C. IBP Bounds Tightness

As highlighted in §IV-C3, IBP is a technique used to accomplish robustness. Researchers studied the tightness of IBP's upper and lower bounds as a metric to determine and formally verify the degree of model robustness against adversarial attacks [118]. A model achieves a provably-guaranteed robustness against an attack if it cannot cross the boundary, no matter how adversaries create adversarial examples [90].

The IBP tightness metric has been utilized in several research works [27], [47], [109], [119]. For example, in [27], Shi *et al.* used the IBP tightness to study the robustness verification

TABLE III
A LISTING OF ROBUSTNESS METRICS. ● MEANS ROBUSTNESS IS SATISFIED, ◐ IS PARTIALLY SATISFIED, AND ○ FOR NOT SATISFIED. THE METRICS WE USE ARE THE ACCURACY RATE, SUCCESS RATE, ERROR RATE, IBP ACCURACY, PERTURBATION SIZE, FAIRNESS, SENSITIVITY, AND DIVERSITY.

Paper	Year	Robustness	Accuracy	Success	Error	IBP	Perturbation	Fairness	Void	Sensitivity	Diversity
Carllini *et al.* [34]	2017	●	✓	✓	✗	✗	✗	✗		✗	✗
Ebrahimi *et al.* [35]	2017	◐	✓	✗	✗	✗	✓	✗		✗	✗
Hosseini *et al.* [103]	2017	◐	✓	✗	✓	✗	✓	✗		✗	✗
Liang *et al.* [104]	2017	○	✓	✗	✓	✗	✓	✗		✗	✗
Wong *et al.* [37]	2017	○	✓	✗	✓	✗	✗	✗		✗	✗
Geiger *et al.* [40]	2018	◐	✓	✓	✓	✗	✗	✗		✗	✗
Glochner *et al.* [41]	2018	○	✓	✗	✗	✗	✓	✗		✓	✓
Iyyer *et al.* [105]	2018	●	✓	✗	✓	✗	✗	✗		✗	✗
Naik *et al.* [106]	2018	◐	✓	✗	✗	✗	✗	✗		✗	✗
Ribeiro *et al.* [107]	2018	●	✓	✗	✗	✗	✗	✗		✗	✗
Braham *et al.* [43]	2019	◐	✓	✗	✗	✗	✓	✗		✗	✗
Chang *et al.* [108]	2019	◐	✗	✗	✗	✗	✗	✓		✗	✗
Cheng *et al.* [26]	2019	●	✗	✓	✗	✗	✗	✗		✗	✗
Eger *et al.* [45]	2019	◐	✓	✗	✗	✗	✗	✗		✗	✗
Haung *et al.* [23]	2019	●	✓	✗	✗	✓	✗	✗		✗	✗
Huang *et al.* [109]	2019	○	✗	✗	✗	✗	✗	✓		✗	✗
Ilyas *et al.* [110]	2019	●	✓	✗	✗	✗	✗	✗		✗	✗
Jia *et al.* [47]	2019	●	✗	✗	✗	✓	✓	✗		✗	✗
Pruthi *et al.* [50]	2019	●	✗	✗	✓	✗	✗	✗		✓	✗
Ren *et al.* [51]	2019	○	✓	✗	✗	✗	✗	✗		✗	✗
Rychalska *et al.* [24]	2019	◐	✓	✗	✗	✗	✗	✗		✗	✗
Zhu *et al.* [111]	2019	◐	✓	✗	✗	✗	✗	✗		✗	✓
Zang *et al.* [53]	2019	○	✗	✓	✗	✗	✗	✗		✗	✗
Cheng *et al.* [54]	2020	○	✗	✓	✗	✗	✓	✗		✗	✗
Dong *et al.* [21]	2020	●	✓	✗	✗	✗	✗	✗		✗	✗
Dong *et al.* [112]	2020	◐	✓	✗	✗	✗	✗	✗		✗	✗
Jin *et al.* [55]	2020	○	✗	✗	✗	✗	✓	✗		✗	✗
Li *et al.* [28]	2020	●	✓	✗	✗	✗	✗	✗		✗	✗
Moriss *et al.* [12]	2020	○	✓	✗	✗	✗	✓	✗		✗	✗
Shi *et al.* [27]	2020	●	✗	✗	✗	✓	✗	✗		✗	✗
Sharma *et al.* [113]	2020	●	✓	✗	✗	✗	✗	✓		✗	✗
Wang *et al.* [56]	2020	●	✗	✗	✗	✗	✗	✗		✗	✗
Wang *et al.* [58]	2020	◐	✓	✗	✗	✗	✗	✗		✗	✓
Ye *et al.* [20]	2020	●	✓	✗	✗	✗	✗	✗		✗	✗
Yoo *et al.* [15]	2020	◐	✗	✓	✗	✗	✓	✗		✗	✗
Zhang *et al.* [59]	2020	◐	✓	✗	✗	✗	✗	✗		✗	✗
Zhout *et al.* [60]	2020	◐	✓	✗	✗	✗	✓	✗		✗	✗
Czarnowska *et al.* [114]	2021	○	✓	✗	✗	✗	✗	✓		✗	✗
Feng *et al.* [14]	2021	●	✓	✗	✗	✗	✗	✗		✓	✓
Goel *et al.* [25]	2021	◐	✓	✗	✗	✗	✗	✗		✗	✗
Xu *et al.* [62]	2021	●	✓	✗	✗	✗	✗	✗		✓	✓
Zeng *et al.* [19]	2021	◐	✗	✓	✗	✗	✓	✗		✗	✗

problem for transformers. In [47], Jia *et al.* used the same metric to study certified robustness to word substitutions and considered an exponentially large family of label-preserving transformations where each word in the input text can be swapped with a similar one. The advantage of using the IBP tightness metric is that it can be used to evaluate verifiable robustness of NLP models to word substitution attacks. On the other hand, this metric alone should not be used as an indication of the true certified robustness. Ideally, other evaluation metrics should be used in conjunction with IBP tightness, such as normal accuracy and training accuracy.

D. Classification Accuracy

The classification accuracy is a simple extension of the accuracy metric, and refers to NLP model's ability to correctly classify input texts under different attack methods (e.g., white-box and black-box attacks, or word-level and character-level substitution attacks, among other settings) [120].

The classification accuracy has been utilized by numerous research works [34], [35], [40], [41], [45], [59], [103], [105], [106]. For example, in [59], Zhang *et al.* used the classification accuracy metric to evaluate their proposed Metropolis-Hastings Sampling Algorithm (MHA) and demonstrated that MHA under classification accuracy outperforms the baseline model on attacking capability.

Similar to other accuracy measures, this metric is easy to calculate and interpret, providing an ideal mean for easily comparing different algorithms. However, on the downside, this metric is agnostic to the quality of the individual examples contributing to the classification accuracy.

E. Diversity

Diversity implies that examples from training data of one class are as differentiable as possible from training data of another class to promote invariance in training data [47].

The diversity metric has been utilized by numerous research studies [26], [43], [44], [111], [120] to measure the classification accuracy of NLP models as part of the robustness to adversarial attacks. For example, in [111], Zhu *et al.* propose a new adversarial training algorithm called FreeLB, which provides a higher invariance in the embedding space by perturbing input words to minimize the resulting adversarial risk on the input text. For validation, they apply their approach to transformer-based models in both NLU and reasoning tasks. Their experiments show that, when applied only to the fine tuning stage, their approach is able to improve the overall test score of BERT-based model from 78.3% to 79.4%, and RoBERTa-large model from 88.5% to 88.8%. The authors, however, stop short in explaining how to measure the run-time of their algorithm because the performance and accuracy of search algorithms sometimes become a trade-off issue. There are many search algorithms in the literature such as genetic algorithm, particle swarm optimization, greedy search, among other. Most of these algorithms have been thoroughly studied and could have been considered for this research as well [15].

One strength of the diversity metric is that it supports a qualitative notion across NLP domains making it suited for evaluating model in ways that are not exhibited in any of the prior metrics. Moreover, this metric allows us to understand how the NLP models generalize in predicting validation target features. However, it is unclear how this metric can be used to measure diversity via precision and recall.

F. Fairness

Fairness in the context of machine learning, and NLP in particular, refers to the fair representation of data points for a particular language understanding task. This metric also aims to ensure that the NLP models do not make erroneous assumptions to produce prejudice results. As a case in point, Amazon?s job candidate NLP-based system was deemed to have prejudice against female applicants because the NLP model did not have a fair representation for female applicants? resumes (i.e., it was not trained with enough female resumes).

The fairness metric has been utilized by various studies [108], [109], [113], [114], [121] to evaluate the degree of NLP robustness and ensure that large models are fair. Sharma *et al.* [113] presented a simple data augmentation technique that selectively adds a subset of synthetic points in order to meet a fairness criterion without compromising the accuracy. Experiments are performed on three datasets where they have shown that their method outperforms prior methods to lessen bias while maintaining the accuracy.

This metric can help reduce and control algorithmic biases and increase how fairly models perform in the real-world. The weakness of this metric is that it is coarse-grained, and cannot easily detect or even differentiate between the unintended biases that may exist in NLP models. Furthermore, it should not be the only measure to determine that whether algorithmic biases are completely removed from models or not.

G. Insights and Open Directions

It is clear that the metrics utilized in the literature for measuring the performance and robustness of NLP models are diverse. This diversity of metrics is necessitated by the variety of shortcomings that some of those metrics have, precluding their use in broad set of applications. Some others are advocated for their simplicity and ease of interpretation.

In general, NLP models have to be evaluated to determine their robustness to adversarial attacks, where different metrics may prove more practical than others. The robustness can be evaluated by computing the upper and lower bounds of IBP, for instance. The upper bound can be minimized using the back-propagation approach. The lower bound can be achieved using IBP to measure certified accuracy [78], [122].

Our analysis of the surveyed work shows that NLP models are more robust to adversarial attacks (e.g., word substitution attacks) when trained with robust training (i.e., the IBP) as opposed to normally-trained models which fared poorly (classification accuracy of 36.0%) under the same adversarial attacks [78]. An interesting area of research in this context is whether models trained with data augmentation would be more robust to attacks than robustly trained models. Another recommendation is to explore and experiment with clean versus robust accuracy metrics. We observed that robust training fared well (classification accuracy of up to 87%) against adversarially perturbed words [78], but we do not know if this robustness would cause any increase or drop in clean accuracy (accuracy on clean/unperturbed words).

Overall, our exploration of the metrics space calls for work in various directions to fill various gaps. (1) While there is a significant initial work on the understanding of bias in the broad NLP community, the metrics developed so far are limited in many ways, and there is a need for developing techniques for assessing fairness and removing biases in data to evaluate how NLP models would perform when deployed to the real-wold. Namely, it would be interesting and worthwhile to extend the existing notions and metrics to techniques that address the existence of residual and unintended biases in datasets. (2) Diversity is understood in terms of the accuracy as a target metric, and it would be worthwhile to extend the diversity metrics to concretely evaluate models using the precision and recall as potential factors. (3) There seems to be a gap and need for techniques to allow the integration and use of IBP tightness metric in conjunction with other metrics such as training accuracy and normal accuracy.

VI. Defense Mechanisms

Neural NLP systems must learn the fragile predictability of natural languages in order to address the generalization flaws of NLP systems [64]. We have already seen how NLP models trained with standard data are vulnerable to adversarial attacks [39]. To address those vulnerabilities, a range of techniques have been studied in the literature, including robustness through data augmentation, adversarial training, and multi-task learning. We note that the literature has several surveys that address each individual technique, which we refer the reader to, although we highlight the high-level ideas of each of those techniques and exemplify the techniques by some sample works for the completeness in treating the subject.

A. Data Augmentation

In the computer vision domain, data augmentation and robust training, as defense mechanisms, have been proven to robustify neural models to adversarial perturbation [36], [123]. Inspired by those developments in the vision field, NLP researchers have considered adversarial training, data augmentation, and robust training as defense mechanisms to robustify NLP models (as depicted in Figure 2). For example, Zeng *et al.* [19] proposed a certifiably robust defense by randomly masking a certain words from the input to defend against both word substitution based attacks and character-level perturbations. The authors claim that they can certify the classifications of over 50% texts to any perturbation of 5 words on AGNEWS dataset, and 2 words in the SST-2 dataset (dataset-dependent). The interested reader could find more details in the comprehensive survey on data augmentation techniques, their advantages, and disadvantages in [14].

B. Adversarial Training

There has been also a significant body of work on the use of adversarial training in defending against attacks on NLP models and systems. It is noted that such attacks are not limited to this application domain, and are prevalent to most learning-based systems.

Given the sufficiency of the prior survey work, and the pace of progress in this domain with respect to the NLP models and applications, the interested reader might refer to the survey of Chakraborty *et al.* for more related works [124].

C. Multi-Task Learning

Multi-Task Learning (MTL) is a learning technique that enables researchers to share useful information or representations between and among related machine learning tasks. This technique is so popular that it has been adopted by researchers and practitioners across many domains including computer vision, speech recognition, and NLP tasks [125]. The benefit of sharing information from related tasks offers the ability to generalize deep learning models more efficiently on the original task. The MTL technique has been utilized in numerous research studies including [126]–[129] and the way this technique is used in those works is very similar. For example, in [126], Tu *et al.* proposed to use multi-task learning (MTL) to improve generalization as a form of robustness in NLP models. The authors experimented on NLI and paraphrase identification to show that MTL leads to significant performance gains. The authors demonstrated the importance of data augmentation and diversity for addressing spurious correlations challenges. The study was carried out on NLI and paraphrase identification (PI).

D. Insights and Open Directions

We observe that data augmentation, adversarial training, multi-task learning, and robust training all have a positive impact on the classification accuracy of NLP models thereby contributing to robustness. We have also noticed from analyzing the literature that robust training outperforms both adversarial training as well as data augmentation when it comes to robustness to adversarial attacks [78].

Overall, our exploration of the defense mechanisms space calls for work in various directions to fill various gaps. (1) While there is a significant initial work on the utilization of adversarial training in the broad NLP community. The adversarial techniques available so far are limited in many ways, and there is a need for developing techniques that evaluate NLP models' robustness beyond deployment to the real-world. Namely, it would be interesting and worthwhile to quantify the impact of adversarial training on rapidly evolving language models as such models will be exposed to unsean data after deployment. (2) There seems to be a gap and the need for robust training techniques to test model robustness across various models because most of the research works in this context conduct robust training under a certain model architecture (e.g., BERT, Glove, etc.). To achieve robustness to adversarial attacks, an NLP model must be evaluated using more than one architecture on various datasets. For instance, in a sentiment analysis task, an NLP model should be evaluated using embeddings such as BoW, GloVe, Word2Vec, and RoBERTa on benchmark datasets from GLUE. (3) Although data augmentation has proven to increase model prediction accuracy, it has not been thoroughly examined to see its long-term impact on model performance. Because language models shift and drift after deployment to the real-work, it is paramount to develop techniques for reevaluating the impact of data augmentation on the long-run. Additionally, most of the research studies that leverage randomized smoothing with IBP, fail to consider the run time aspect, which is the overhead incurred during computation. We wish to motivate the NLP research community to consider the above gaps as future research directions.

VII. Embedding Techniques

Several embedding techniques have been utilized in the literature for representing text and natural language entities (in Figure 2). The choice of those models is influenced by various factors, including their fitness to the studied applications, performance, and robustness. In the following, we review some of those embeddings and where they are used. A summary and contrast of some of those works are shown in Table IV.

A. Representation Techniques

1) Bag of Words: The bag-of-words model, or BoW for short, is a technique used for extracting features from raw data for use in NLP models [132]. Moreover, this technique is considered as a popular text embedding technique widely used for text classification tasks such as sentiment analysis. In the BoW technique, the input text is represented as the bag of its words without regard to word order or grammar.

The BoW embedding method has been implemented by numerous research studies [40], [60], [64], [75], [106] to achieve robustness for NLP tasks such a sentiment analysis and spam filtering. For example, in [40], Geiger *et al.* proposed a method for generating semantically challenging NLI data sets using the popular BoW embedding technique and showed that a range of NLI neural models (especially models based on the BoW embedding technique) invariably learn sub-optimal solutions and fail to encode crucial information. The authors concluded that certain NLP models are not fit for certain NLP tasks due to inherent weaknesses in their underlying architecture (hence the implementation of BoW).

TABLE IV

REPRESENTATIVE LITERATURE WORK WITH VARIOUS EMBEDDING TECHNIQUES. ● STANDS FOR ROBUSTNESS BEING SATISFIED, ◐ FOR PARTIALLY SATISFIED, AND ○ FOR NOT SATISFIED.

Paper	Year	Robustness	BoW	Word2Vec	Glove	ELMO	ROBERTA	BERT	LSTM
Ebrahimi *et al.* [35]	2017	◐	✗	✗	✗	✗	✗	✗	✓
Hoseini *et al.* [103]	2017	◐	✗	✓	✗	✓	✗	✓	✗
Jia *et al.* [78]	2017	●	✗	✗	✗	✗	✗	✓	✓
Wong *et al.* [37]	2017	○	✗	✗	✗	✗	✗	✗	✓
Zhao *et al.* [38]	2017	○	✗	✗	✗	✗	✗	✗	✓
Alzantot *et al.* [39]	2018	○	✗	✗	✓	✗	✗	✓	✗
Gao *et al.* [75]	2018	◐	✓	✗	✓	✗	✗	✗	✗
Geiger *et al.* [40]	2018	○	✓	✗	✗	✗	✗	✗	✓
Gong *et al.* [42]	2018	○	✗	✗	✓	✗	✗	✗	✗
Naik *et al.* [106]	2018	◐	✓	✗	✗	✗	✗	✗	✓
Braham *et al.* [43]	2019	◐	✗	✗	✗	✗	✗	✗	✓
Cheng *et al.* [26]	2019	●	✗	✗	✗	✗	✗	✓	✗
Haung *et al.* [23]	2019	●	✗	✗	✗	✗	✗	✓	✗
Jia *et al.* [47]	2019	●	✓	✗	✗	✗	✗	✗	✓
Pruthi *et al.* [50]	2019	●	✗	✗	✗	✗	✗	✓	✓
Ren *et al.* [51]	2019	○	✗	✗	✓	✗	✗	✗	✓
Rychalska *et al.* [24]	2019	◐	✗	✗	✓	✗	✗	✓	✗
Schmitt *et al.* [130]	2019	○	✗	✓	✗	✗	✗	✗	✗
Wallace *et al.* [131]	2019	●	✗	✓	✗	✓	✗	✗	✗
Zang *et al.* [53]	2019	○	✗	✗	✓	✗	✗	✓	✓
Zhu *et al.* [111]	2019	◐	✗	✗	✗	✗	✓	✓	✗
Dong *et al.* [21]	2020	●	✗	✗	✗	✗	✗	✗	✓
Gardner *et al.* [116]	2020	○	✗	✗	✗	✓	✓	✓	✗
Hendrycks *et al.* [31]	2020	◐	✗	✓	✓	✗	✓	✓	✗
Jin *et al.* [55]	2020	○	✗	✗	✗	✗	✗	✓	✗
Moriss *et al.* [12]	2020	○	✗	✗	✗	✗	✗	✓	✓
Shi *et al.* [27]	2020	●	✗	✗	✗	✗	✗	✓	
Wang *et al.* [56]	2020	●	✗	✗	✗	✗	✓	✓	✗
Ye *et al.* [20]	2020	●	✗	✗	✗	✗	✗	✓	✗
Yoo *et al.* [15]	2020	●	✗	✗	✓	✗	✗	✓	✓
Zhang *et al.* [59]	2020	◐	✗	✗	✗	✗	✗	✓	✓
Zhout *et al.* [60]	2020	◐	✓	✗	✗	✗	✗	✗	✓
Fabbri *et al.* [30]	2021	●	✗	✗	✗	✗	✗	✓	✗
Goel *et al.* [25]	2021	◐	✗	✗	✗	✗	✗	✓	✗
Schiller *et al.* [61]	2021	●	✗	✗	✗	✗	✗	✓	✗
Xu *et al.* [62]	2021	●	✗	✗	✗	✗	✗	✓	✓
Zeng *et al.* [19]	2021	◐	✗	✗	✗	✗	✓	✓	✗

The strengths of the BoW technique are that: (1) it is easy to implement. (2) it offers flexibility and ability of customizing it to the specific task, text data type, and structure. However, on the downside, and because BoW ignores words ordering, it will lead to ignoring the context, which negatively impacts word meanings (semantics). The fact that word meaning and context are ignored by BoW, is a significant limitation of the model's linguistic capabilities.

2) Word2Vec: Word2Vec is an model used to transform words into vectors. This is achieved by representing text into a numerical format that deep neural networks can understand [133]. This step is necessary to help NLP models understand text in the form of numbers. Word2vec (if provided with sufficient data and context) has the capability to make highly accurate predictions for tasks such as sentiment analysis (i.e., by classifying reviews with high certainty) and topic modeling.

The Word2Vec technique has been utilized by numerous research studies including [31], [103], [130], [131] to reveal a model's weaknesses by showing a lower classification accuracy score. For instance, in [131], Wallace *et al.* introduced triggers as a new form of universal adversarial perturbation and used Word2Vec to evaluate the robustness of NLP models to adversarial attacks. The authors proposed a gradient guided search over tokens to finds trigger sequences that successfully lead to the target prediction. Their experiments utilized Word2Vec embedding technique and demonstrated a sharp decrease in accuracy of certain NLP models. The benefit of utilizing the Word2Vec representation technique is that it is simple and intuitive. Moreover, the way it is implemented enables the Word2Vec to easily learn how words are represented in classification tasks. Another advantage of this technique is that it does not require huge preprocessing, and hence less memory, because the model accepts data in an online manner. On the downside, this technique suffers from its inability to deal with out-of-vocabulary words, since Word2Vec model is unable to interpret unseen words. Moreover, this technique does not scale to other languages because it would require new embedding metrics.

3) GloVe: Global Vector for word representation is another representative learning technique which is a reformulation of the Word2vec optimization algorithm [134]. In this method, words are represented as vectors to create word co-occurrence matrices [135]. The core concept in this method is to determine the frequency of word occurrence in the co-occurrence matrix.

The GloVe learning technique has been utilized by numerous research studies [24], [31], [39], [42], [51] to study the impact of word learning representation on robustness. For instance, in [24], Rychalska *et al.* introduced WildNLP, a framework for studying model stability with text corruptions, e.g., keyboard errors and misspelling. In their pursuit, the authors use the GloVe embedding technique as a baseline to evaluate NLP robustness. One advantage of this learning technique is that it considers the frequency of co-occurrences (also refereed to as global statistics) crucial to building word embeddings. Thus word re-occurrences is directly tied to word vectors.

Because GloVe is considered a count-based model, it potentially requires more computational power for more processing, a major draw-back of this technique.

4) ELMO: Embedding from Language Models, or ELMO, is a word embedding technique for transforming a sequence of words into a sequence of vectors. In this approach, the input data is represented as character-level tokens and the output is word-level embeddings [136]. We note that this learning technique differs from the previously mentioned learning techniques in that it computes vectors for an entire sentence instead of assigning a vector for each word embedding. This means that the same word can be assigned to different word vectors if the context is different. This is where the difference between ELMO and other traditional embedding techniques (e.g., GloVe, and Word2vec) come into play. ELMO has been successfully utilized in numerous works, including [49], [103], [116], [131], to accomplish high classification accuracy in the context of robustness for many NLP linguistic tasks including text classification and question-answering.

One advantage of ELMO is that it can handle the out-of-distribution issue because of its ability to use character embeddings to represent word embeddings. However, on the downside, ELMO requires huge computation time to realize the word vectors [137], a limitation we may overcome by pre-computing the vectors offline.

5) Fastext: This technique is an extension of the Word2Vec method: instead of transforming each word from the input text into a vector as output, this method represents each word as an n-gram of characters for the output. Fastext can be used to reveal a model's "blind spots" in the context of robustness and has been utilized for such purpose by numerous research studies [107], [138]–[141]. For instance, in [107], Ribeiro *et al.* conducted a study to find bugs in NLP models based on Fastext and presented a semantically equivalent adversaries and semantic-preserving perturbations, defined as perturbations that produce changes in the model's predictions. The authors implemented their method using several NLP tasks; QA and SA.

The benefit of using the Fastext learning technique is that it employs a simple and efficient baseline for sentence classification and uses N-gram features to reduce computation time and enhance efficiency. However, it might take a longer time to train a Fastext model because of the fact it uses N-gram features which could be greater than the number of words. This means that Fastext embedding is fit for only a certain linguistic tasks such as classification tasks.

B. Insights and Open Directions

It is evident that representation learning techniques utilized in the literature for evaluating NLP models are diverse. Impressive results on a certain learning technique should not imply that a given NLP model will perform equally well when deployed in the real-world. Overall, our exploration of those techniques calls for work in various directions to fill various gaps. (1) While there is a significant initial work on the utilization of various embedding techniques in the broad NLP community, the current research works fall short in using the same embedding technique/model (e.g., Fastext, BERT) across various linguistics tasks. (2) To harness the full power and potential of an embedding technique, there is a need for developing learning techniques, e.g., it would be interesting and worthwhile to build an encoding layer (within the model's architecture) that can enforce resistance to perturbations. (3) There seems to be a gap in exploring the reuse of encoding across various linguistics tasks for robustness, which has the potential to improve classification accuracy.

VIII. Robustness via Benchmark Datasets

NLP systems have performed remarkably on a wide spectrum of linguistic tasks due, in large, to the emergence of deep neural networks and unsupervised pre-training [142]. Standard benchmark datasets have accomplished excellent results across many NLP tasks. For example, the SQuAD dataset has 95.4% F-1 score which outperforms human accuracy [143].

As tempting as it may be to believe how well NLP models perform on standard datasets generally, often these models are actually solving the "dataset problem" rather than solving the underlying task with sufficient generalizations. For instance, in [143], a BERT model is trained on the SQuAD dataset and achieved an impressive 86.5% F-1 score. We note, however, that these results are chiefly because the model is tested on a data that is created in the same way as the training data, following the same distribution and patterns. This, in turn, provides a false-sense of confidence in NLP model performance that may generalize for OOD samples. On the flip side, when the same model was tested on TriviaQA dataset (which is created in the same format as the SQuAD dataset), the F-1 score dropped to 35.6%.

It is important to understand the data utilized in validating and evaluating a task to reach conclusions on generalization. One such best practice to achieve this goal is to challenge the model with OOD samples. For instance, one approach to show this practice is by exposing the model to a training set, and not to samples from within the distribution of the testing set. One possible benefit of examining models with such settings is that surface cues can be identified to show limitations of models, as pointed out in a previous section, where gaps could be addressed by using techniques that can help with robustness and generalization, e.g., data augmentation techniques [14].

Another approach to extend the generalization of models and improve the robustness to weak forms of attacks is to utilize widely-accepted and standard benchmark dataset: the fact that those datasets are standard imply that they went through rigorous evaluation for representation and soundness of collection. For instance, GLUE (General Language Understanding Evaluation), depicted in Figure 2, is one of evaluation tools, which is a collection of nine benchmark datasets. GLUE is designed for analyzing and effectively evaluating NLP systems on three linguistic tasks, classification (benchmark datasets: SST-2 and CoLA), paraphrasing (benchmark datasets: MRPC, STS-B, and QQP), and inference (benchmark datasets: MNLI, WNLI, QNLI, and RTE) [142].

While benchmarks are an excellent way to improve the robustness and examine the generalization of models through well-vetted data, their main purpose is to mitigate bias and address spurious correlations, which we review in the following.

A. Dataset Bias

One of the benefits of utilizing rich benchmarks is addressing the explicit bias. Dataset bias in NLP refers to a form of error in which certain elements (i.e., variables or attributes) of a dataset are more heavily represented than others [144], this skewing the resulting NLP model recognizing such bias and affecting its operation against a more diverse data distribution. In other words, the biased dataset does not accurately reflect a model's true use case in the real-world, resulting in analytical errors and misleading classification accuracy levels.

The issue of dataset bias in NLP has been studied extensively in numerous research works [44], [46], [144], [145]. The way those research works examine data bias varies depending on the task and domain. For example, in [46], He *et al.* argued that NLI tasks are susceptible to learning dataset bias via surface cues; superficial cues that are associated with the label on a particular dataset. They investigated a recently proposed approach, called FLite [146]. FLite adversarially filters dataset biases to mitigate the prevalent overestimation and overfitting of data in models. The authors demonstrate that FLite significantly reduces the measurable dataset biases, where models trained on the filtered datasets yielded better generalization to OOD tasks. However, their study stops short of extending its applicability to other language tasks, e.g., sentiment analysis, spam detection.

Clark *et al.* [44] argued that NLP models suffer from generalization and OOD issues due to biases in training datasets. Their study showed that the prior knowledge of this biases will enable training a model to be more robust to domain shift. The study demonstrated through experiments on several datasets with out-of-domain test sets huge robustness gains.

Finally, Schiller *et al.* [61] introduced a stance detection benchmark, called StD, to add and evaluate adversarial attack sets for NLP tasks. Their study demonstrated that the existence of biases inherited from multiple datasets by design leads to lack of robustness against adversarial examples. The authors stressed the need to focus on robustness and de-biasing strategies in multi-task learning approaches. However, the study did not offer recommendations on the applicability of this approach to other NLP tasks such as sentiment analysis, NLI, and question answering models.

B. Spurious Correlation

NLP models are generally prone to learning surface cues from training data which leads to the phenomena "Spurious correlation". Spurious correlations fool NLP models into making wrong predictions because models tend to rely on simple shortcuts rather than relying on the actual, typically complex, relationships in making such predictions.

The spurious correlation issue has been examined extensively in numerous research studies including the works in [52], [57], [147]–[149]. The way those research works tackle this issue varies greatly, depending on the linguistic task. For example, in [52], Yaghoobzadeh *et al.* presented a novel approach to design more robust NLP models and address the spurious correlation issue systematically. Their framework is based on *example forgetting*, where they find minority examples without any knowledge of the correlations present in the dataset. They tested their technique using three NLP taks (NLI, paraphrase identification, and fact verification) and showed consistent robustness gains. However, the study stops short of discussing other strategies to address the harm of spurious correlations such as data diversity and invariance.

Kaushik *et al.* [49] studied whether NLP models pick up spurious patterns (e.g., if they are taking short-cuts instead of learning about the dataset when making predictions). They discovered that BERT and BiLSTM models trained on original data fail to make correct predictions, while performing remarkably better when trained on combined datasets (counterfactually-revised counterparts). They have shown the results to generalize for multiple tasks; i.e., sentiment analysis and NLI. In their pursuit, the authors used humans in the loop to provide labels (predictions) and to intervene upon the data which may not be realistic given large datasets in the real-world. The study, however, did not offer any insights on how their models would perform on challenging adversarial datasets where spurious correlations do not necessarily hold.

Wang *et al.* [57] examined the effect of spurious correlation on the accuracy and robustness of text classifiers using a BERT model. They argue that NLP models are prone to learning surface cues during training, which may cause models to make incorrect predictions. They conducted studies using sentiment analysis NLP tasks. The study suggests feature engineering strategies to accomplish robustness, although it stops shorts of offering any insights on the transferibility and generalization of their approach to other architectures, e.g., CNN.

C. Insights and Open Directions

It is clear that rich benchmark datasets utilized in the literature for evaluating NLP models are diverse but inconsistent. Impressive results on benchmark datasets should not imply that a given NLP model will perform remarkably when deployed in the real-world. Research has shown that models are susceptible to solving the dataset rather than solving the underlying language understanding task [48].

Overall, our exploration of the benchmark datasets space calls for work in various directions to fill various gaps. (1) While there is a significant initial work on the utilization of benchmark datasets that are free of bias in the broad NLP community, the datasets available so far are limited in many ways, and there is a need for developing techniques for identifying and removing biases in data to evaluate how

NLP models would perform when deployed to the real-world. Namely, it would be interesting and worthwhile to extend the existing notions and benchmarks to techniques that address the existence of spurious correlation in datasets. (2) Using the same dataset for both training and testing might provide a false-sense of a model's robustness and accuracy. It would be interesting and worthwhile to test NLP models on various datasets by adopting the super GLUE, which is a successor of GLUE, a more challenging suite of datasets for various linguistic tasks. (3) There seems to be a gap and the need for a unified evaluation framework to enable comprehensive evaluations across various linguistic tasks in a fair and reproducible fashion. Namely, developing a standardized, unified evaluation benchmark dataset would be intriguing. In addition, modeling a strong set of baselines to be used as a test bed and trained on domain-specific data would also be interesting.

IX. Conclusion

This paper presents a survey on NLP robustness research in a consistent and systemic way. We identified various gaps in the literature with recommendations on future area of research directions following various elements in the NLP pipeline. As numerous real-world NLP projects have failed after deployment due to lack of robustness, exploring the robustness as a multi-dimensional concept that requires the development of new techniques is paramount. We note that newly developed techniques should address the spurious correlation challenges and achieve high out-of-distribution accuracy to ensure sufficient sensitivity to perturbations and ultimately lead to high precision in realistic text classification settings. Overall, it is our hope that this research work will serve as a fresh guide for the research community on technique, metric, and dataset to use, and motivate for additional interest and work in this space addressing the various gaps.

References

[1] S. Mannarswamy and S. Roy, "Evolving ai from research to real life-some challenges and suggestions." in *IJCAI*, 2018, pp. 5172–5179.

[2] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, "Natural language processing: an introduction," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 544–551, 2011.

[3] B. Liu, *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge university press, 2020.

[4] A. Abusnaina, A. Khormali, H. Alasmary, J. Park, A. Anwar, and A. Mohaisen, "Adversarial learning attacks on graph-based iot malware detection systems," in *International Conference on Distributed Computing Systems (ICDCS)*, 2019.

[5] N. Zhang, X. Mi, X. Feng, X. Wang, Y. Tian, and F. Qian, "Understanding and mitigating the security risks of voice-controlled third-party skills on amazon alexa and google home," *arXiv preprint arXiv:1805.01525*, 2018.

[6] L. Blue, L. Vargas, and P. Traynor, "Hello, is it me you're looking for? differentiating between human and electronic speakers for voice interface security," in *ACM Conference on Security & Privacy in Wireless and Mobile Networks*, 2018, pp. 123–133.

[7] S. Chen, K. Ren, S. Piao, C. Wang, Q. Wang, J. Weng, L. Su, and A. Mohaisen, "You can hear but you cannot steal: Defending against voice impersonation attacks on smartphones," in *International Conference on Distributed Computing Systems (ICDCS)*, 2017.

[8] M. Abuhamad, A. Abusnaina, D. Nyang, and D. Mohaisen, "Sensor-based continuous authentication of smartphones' users using behavioral biometrics: A contemporary survey," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 65–84, 2020.

[9] A. Abusnaina, A. Khormali, D. Nyang, M. Yuksel, and A. Mohaisen, "Examining the robustness of learning-based ddos detection in software defined networks," in *2019 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE, 2019, pp. 1–8.

[10] H. Alasmary, A. Abusnaina, R. Jang, M. Abuhamad, A. Anwar, D. Nyang, and D. Mohaisen, "Soteria: Detecting adversarial examples in control flow graph-based malware classifiers," in *International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2020, pp. 888–898.

[11] E. Buber, B. Dırı, and O. K. Sahingoz, "Detecting phishing attacks from url by using nlp techniques," in *2017 International conference on computer science and Engineering (UBMK)*. IEEE, 2017, pp. 337–342.

[12] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi, "Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp," *arXiv preprint arXiv:2005.05909*, 2020.

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[14] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy, "A survey of data augmentation approaches for nlp," *arXiv preprint arXiv:2105.03075*, 2021.

[15] J. Y. Yoo, J. X. Morris, E. Lifland, and Y. Qi, "Searching for a search method: Benchmarking search algorithms for generating nlp adversarial examples," *arXiv preprint arXiv:2009.06368*, 2020.

[16] J. Li, T. Tang, W. X. Zhao, and J.-R. Wen, "Pretrained language models for text generation: A survey," *arXiv preprint arXiv:2105.10311*, 2021.

[17] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, "Adversarial attacks on deep-learning models in natural language processing: A survey," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 3, pp. 1–41, 2020.

[18] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, "Beyond accuracy: Behavioral testing of nlp models with checklist," *arXiv preprint arXiv:2005.04118*, 2020.

[19] J. Zeng, X. Zheng, J. Xu, L. Li, L. Yuan, and X. Huang, "Certified robustness to text adversarial attacks by randomized [mask]," *arXiv preprint arXiv:2105.03743*, 2021.

[20] M. Ye, C. Gong, and Q. Liu, "Safer: A structure-free approach for certified robustness to adversarial word substitutions," *arXiv preprint arXiv:2005.14424*, 2020.

[21] X. Dong, A. T. Luu, R. Ji, and H. Liu, "Towards robustness against natural language word substitutions," in *International Conference on Learning Representations*, 2020.

[22] A. Kumar, A. Levine, S. Feizi, and T. Goldstein, "Certifying confidence via randomized smoothing," *arXiv preprint arXiv:2009.08061*, 2020.

[23] P.-S. Huang, R. Stanforth, J. Welbl, C. Dyer, D. Yogatama, S. Gowal, K. Dvijotham, and P. Kohli, "Achieving verified robustness to symbol substitutions via interval bound propagation," *arXiv preprint arXiv:1909.01492*, 2019.

[24] B. Rychalska, D. Basaj, A. Gosiewska, and P. Biecek, "Models in the wild: On corruption robustness of neural nlp systems," in *International Conference on Neural Information Processing*. Springer, 2019, pp. 235–247.

[25] K. Goel, N. Rajani, J. Vig, S. Tan, J. Wu, S. Zheng, C. Xiong, M. Bansal, and C. Ré, "Robustness gym: Unifying the nlp evaluation landscape," *arXiv preprint arXiv:2101.04840*, 2021.

[26] M. Cheng, W. Wei, and C.-J. Hsieh, "Evaluating and enhancing the robustness of dialogue systems: A case study on a negotiation agent," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 3325–3335.

[27] Z. Shi, H. Zhang, K.-W. Chang, M. Huang, and C.-J. Hsieh, "Robustness verification for transformers," *arXiv preprint arXiv:2002.06622*, 2020.

[28] J. Li, T. Du, S. Ji, R. Zhang, Q. Lu, M. Yang, and T. Wang, "Textshield: Robust text classification based on multimodal embedding and neural machine translation," in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020, pp. 1381–1398.

[29] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *arXiv preprint arXiv:2106.04554*, 2021.

[30] A. R. Fabbri, W. Kryściński, B. McCann, C. Xiong, R. Socher, and D. Radev, "Summeval: Re-evaluating summarization evaluation," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 391–409, 2021.

[31] D. Hendrycks, X. Liu, E. Wallace, A. Dziedzic, R. Krishnan, and D. Song, "Pretrained transformers improve out-of-distribution robustness," *arXiv preprint arXiv:2004.06100*, 2020.

[32] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[33] B. Jang, M. Kim, G. Harerimana, S.-u. Kang, and J. W. Kim, "Bi-lstm model to increase accuracy in text classification: Combining word2vec cnn and attention mechanism," *Applied Sciences*, vol. 10, no. 17, p. 5841, 2020.

[34] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. IEEE, 2017, pp. 39–57.

[35] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "Hotflip: White-box adversarial examples for text classification," *arXiv preprint arXiv:1712.06751*, 2017.

[36] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[37] C. Wong, "Dancin seq2seq: Fooling text classifiers with adversarial text example generation," *arXiv preprint arXiv:1712.05419*, 2017.

[38] Z. Zhao, D. Dua, and S. Singh, "Generating natural adversarial examples," *arXiv preprint arXiv:1710.11342*, 2017.

[39] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang, "Generating natural language adversarial examples," *arXiv preprint arXiv:1804.07998*, 2018.

[40] A. Geiger, I. Cases, L. Karttunen, and C. Potts, "Stress-testing neural models of natural language inference with multiply-quantified sentences," *arXiv preprint arXiv:1810.13033*, 2018.

[41] M. Glockner, V. Shwartz, and Y. Goldberg, "Breaking nli systems with sentences that require simple lexical inferences," *arXiv preprint arXiv:1805.02266*, 2018.

[42] Z. Gong, W. Wang, B. Li, D. Song, and W.-S. Ku, "Adversarial texts with gradient methods," *arXiv preprint arXiv:1801.07175*, 2018.

[43] S. Barham and S. Feizi, "Interpretable adversarial training for text," *arXiv preprint arXiv:1905.12864*, 2019.

[44] C. Clark, M. Yatskar, and L. Zettlemoyer, "Don't take the easy way out: Ensemble based methods for avoiding known dataset biases," *arXiv preprint arXiv:1909.03683*, 2019.

[45] S. Eger, G. G. Şahin, A. Rücklé, J.-U. Lee, C. Schulz, M. Mesgar, K. Swarnkar, E. Simpson, and I. Gurevych, "Text processing like humans do: Visually attacking and shielding nlp systems," *arXiv preprint arXiv:1903.11508*, 2019.

[46] H. He, S. Zha, and H. Wang, "Unlearn dataset bias in natural language inference by fitting the residual," *arXiv preprint arXiv:1908.10763*, 2019.

[47] R. Jia, A. Raghunathan, K. Göksel, and P. Liang, "Certified robustness to adversarial word substitutions," *arXiv preprint arXiv:1909.00986*, 2019.

[48] T. Jung, D. Kang, L. Mentch, and E. Hovy, "Earlier isn't always better: Sub-aspect analysis on corpus and system biases in summarization," *arXiv preprint arXiv:1908.11723*, 2019.

[49] D. Kaushik, E. Hovy, and Z. C. Lipton, "Learning the difference that makes a difference with counterfactually-augmented data," *arXiv preprint arXiv:1909.12434*, 2019.

[50] D. Pruthi, B. Dhingra, and Z. C. Lipton, "Combating adversarial misspellings with robust word recognition," *arXiv preprint arXiv:1905.11268*, 2019.

[51] S. Ren, Y. Deng, K. He, and W. Che, "Generating natural language adversarial examples through probability weighted word saliency," in *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, pp. 1085–1097.

[52] Y. Yaghoobzadeh, S. Mehri, R. Tachet, T. J. Hazen, and A. Sordoni, "Increasing robustness to spurious correlations using forgettable examples," *arXiv preprint arXiv:1911.03861*, 2019.

[53] Y. Zang, F. Qi, C. Yang, Z. Liu, M. Zhang, Q. Liu, and M. Sun, "Word-level textual adversarial attacking as combinatorial optimization," *arXiv preprint arXiv:1910.12196*, 2019.

[54] M. Cheng, J. Yi, P.-Y. Chen, H. Zhang, and C.-J. Hsieh, "Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 3601–3608.

[55] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is BERT really robust? a strong baseline for natural language attack on text classification and entailment," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 8018–8025.

[56] B. Wang, S. Wang, Y. Cheng, Z. Gan, R. Jia, B. Li, and J. Liu, "Infobert: Improving robustness of language models from an information theoretic perspective," *arXiv preprint arXiv:2010.02329*, 2020.

[57] Z. Wang and A. Culotta, "Identifying spurious correlations for robust text classification," *arXiv preprint arXiv:2010.02458*, 2020.

[58] T. Wang, X. Wang, Y. Qin, B. Packer, K. Li, J. Chen, A. Beutel, and E. Chi, "Cat-gen: Improving robustness in nlp models via controlled adversarial text generation," *arXiv preprint arXiv:2010.02338*, 2020.

[59] H. Zhang, H. Zhou, N. Miao, and L. Li, "Generating fluent adversarial examples for natural languages," *arXiv preprint arXiv:2007.06174*, 2020.

[60] Y. Zhou, X. Zheng, C.-J. Hsieh, K.-w. Chang, and X. Huang, "Defense against adversarial attacks in nlp via dirichlet neighborhood ensemble," *arXiv preprint arXiv:2006.11627*, 2020.

[61] B. Schiller, J. Daxenberger, and I. Gurevych, "Stance detection benchmark: How robust is your stance detection?" *KI-Künstliche Intelligenz*, pp. 1–13, 2021.

[62] Y. Xu, X. Zhong, A. J. Yepes, and J. H. Lau, "Grey-box adversarial attack and defence for sentiment classification," *arXiv preprint arXiv:2103.11576*, 2021.

[63] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1310–1320.

[64] R. Jia, *Building Robust Natural Language Processing Systems*. Stanford University, 2020.

[65] S. Gowal, K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. Mann, and P. Kohli, "On the effectiveness of interval bound propagation for training verifiably robust models," *arXiv preprint arXiv:1810.12715*, 2018.

[66] W. Ryou, J. Chen, M. Balunovic, G. Singh, A. Dan, and M. Vechev, "Fast and effective robustness certification for recurrent neural networks," *arXiv preprint arXiv:2005.13300*, 2020.

[67] I.-F. Kenmogne, V. Drevelle, and E. Marchand, "Interval-based cooperative uavs pose domain characterization from images and ranges," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6349–6356.

[68] A. Sridhar, C. Sitawarin, and D. Wagner, "Mitigating adversarial training instability with batch normalization," in *Proceedings of International Conference on Learning Representation Workshop on Security and Safety in Machine Learning Systems*, 2021.

[69] Z. Lyu, M. Guo, T. Wu, G. Xu, K. Zhang, and D. Lin, "Towards evaluating and training verifiably robust neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4308–4317.

[70] X. Dong, A. T. Luu, R. Ji, and H. Liu, "Towards robustness against natural language word substitutions," *arXiv preprint arXiv:2107.13541*, 2021.

[71] M. Moradi and M. Samwald, "Evaluating the robustness of neural language models to input perturbations," *arXiv preprint arXiv:2108.12237*, 2021.

[72] T. Cui, J. Xiao, L. Li, X. Jiang, and Q. Liu, "An approach to improve robustness of nlp systems against asr errors," *arXiv preprint arXiv:2103.13610*, 2021.

[73] T. Gui, X. Wang, Q. Zhang, Q. Liu, Y. Zou, X. Zhou, R. Zheng, C. Zhang, Q. Wu, J. Ye *et al.*, "Textflint: Unified multilingual robustness evaluation toolkit for natural language processing," *arXiv preprint arXiv:2103.11441*, 2021.

[74] G. Bernier-Colborne and P. Langlais, "Hardeval: Focusing on challenging tokens to assess robustness of ner," in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020, pp. 1704–1711.

[75] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 50–56.

[76] O. Blum, B. Brattoli, and B. Ommer, "X-gan: Improving generative adversarial networks with convex combinations," in *German Conference on Pattern Recognition*. Springer, 2018, pp. 199–214.

[77] D. Szeghy, Z. A. Milacski, A. Fóthi, and A. Lorincz, "Adversarial perturbation stability of the layered group basis pursuit," *def*, vol. 1, p. 2, 2021.

[78] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," *arXiv preprint arXiv:1707.07328*, 2017.

[79] L. Yuan, J. Zeng, and X. Zheng, "Sparsegan: Sparse generative adversarial network for text generation," *arXiv preprint arXiv:2103.11578*, 2021.

[80] T. Tsiligkaridis and J. Roberts, "Understanding frank-wolfe adversarial training," *arXiv preprint arXiv:2012.12368*, 2020.

[81] Ş. İ. Birbil, S.-C. Fang, and R.-L. Sheu, "On the convergence of a population-based global optimization algorithm," *Journal of global optimization*, vol. 30, no. 2-3, pp. 301–318, 2004.

[82] A. Khormali, D. Nyang, and D. Mohaisen, "Generating adversarial examples with an optimized quality," *arXiv preprint arXiv:2007.00146*, 2020.

[83] R. Maheshwary, S. Maheshwary, and V. Pudi, "Generating natural language attacks in a hard label black box setting," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 2021.

[84] X. Li, L. Chen, and D. Wu, "Turning attacks into protection: Social media privacy protection using adversarial attacks," in *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM, 2021, pp. 208–216.

[85] L. Hui, Z. Bo, H. Linquan, G. Jiabao, and L. Yifan, "Foolchecker: A platform to evaluate the robustness of images against adversarial attacks," *Neurocomputing*, vol. 412, pp. 216–225, 2020.

[86] H. Larijani, N. Mtetwa, M. Yousefi, A. Javed *et al.*, "An adversarial attack detection paradigm with swarm optimization," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–7.

[87] K. D. Gupta, Z. Akhtar, and D. Dasgupta, "Determining sequence of image processing technique (ipt) to detect adversarial attacks," *arXiv preprint arXiv:2007.00337*, 2020.

[88] J. Jasser and I. Garibay, "Resilience from diversity: Population-based approach to harden models against adversarial attacks," *arXiv preprint arXiv:2111.10272*, 2021.

[89] T. Suzuki, S. Takeshita, and S. Ono, "Adversarial example generation using evolutionary multi-objective optimization," in *2019 IEEE Congress on evolutionary computation (CEC)*. IEEE, 2019, pp. 2136–2144.

[90] W. Wang, R. Wang, L. Wang, Z. Wang, and A. Ye, "Towards a robust deep neural network in texts: A survey," *arXiv preprint arXiv:1902.07285*, 2019.

[91] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of artificial intelligence research*, vol. 22, pp. 457–479, 2004.

[92] G. Ma, L. Shi, and Z. Guan, "Adversarial text generation via probability determined word saliency," in *International Conference on Machine Learning for Cyber Security*. Springer, 2020, pp. 562–571.

[93] R. Pasunuru and M. Bansal, "Multi-reward reinforced summarization with saliency and entailment," *arXiv preprint arXiv:1804.06451*, 2018.

[94] S. Ding, H. Xu, and P. Koehn, "Saliency-driven word alignment interpretation for neural machine translation," *arXiv preprint arXiv:1906.10282*, 2019.

[95] J. Jeon and M. Kim, "Discovering latent topics with saliency-weighted lda for image scene understanding," *IEEE MultiMedia*, vol. 26, no. 3, pp. 56–68, 2018.

[96] Y. Zang, B. Hou, F. Qi, Z. Liu, X. Meng, and M. Sun, "Learning to attack: Towards textual adversarial attacking in real-world situations," *arXiv preprint arXiv:2009.09192*, 2020.

[97] T. Roth, Y. Gao, A. Abuadbba, S. Nepal, and W. Liu, "Token-modification adversarial attacks for natural language processing: A survey," *arXiv preprint arXiv:2103.00676*, 2021.

[98] F. Qi, C. Yang, Z. Liu, Q. Dong, M. Sun, and Z. Dong, "Openhownet: An open sememe-based lexical knowledge base," *arXiv preprint arXiv:1901.09957*, 2019.

[99] B. Alshemali and J. Kalita, "Generalization to mitigate synonym substitution attacks," in *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 2020, pp. 20–28.

[100] H. Wang, G. Li, X. Liu, and L. Lin, "A hamiltonian monte carlo method for probabilistic adversarial attack and learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[101] Y. Liu, P. Kothari, and A. Alahi, "Collaborative sampling in generative adversarial networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 4948–4956.

[102] S. Garg and G. Ramakrishnan, "Bae: Bert-based adversarial examples for text classification," *arXiv preprint arXiv:2004.01970*, 2020.

[103] H. Hosseini, S. Kannan, B. Zhang, and R. Poovendran, "Deceiving google's perspective api built for detecting toxic comments," *arXiv preprint arXiv:1702.08138*, 2017.

[104] B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi, "Deep text classification can be fooled," *arXiv preprint arXiv:1704.08006*, 2017.

[105] M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer, "Adversarial example generation with syntactically controlled paraphrase networks," *arXiv preprint arXiv:1804.06059*, 2018.

[106] A. Naik, A. Ravichander, N. Sadeh, C. Rose, and G. Neubig, "Stress test evaluation for natural language inference," *arXiv preprint arXiv:1806.00692*, 2018.

[107] M. T. Ribeiro, S. Singh, and C. Guestrin, "Semantically equivalent adversarial rules for debugging nlp models," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 856–865.

[108] K.-W. Chang, V. Prabhakaran, and V. Ordonez, "Bias and fairness in natural language processing," 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): Tutorial Abstracts*, 2019.

[109] P.-S. Huang, H. Zhang, R. Jiang, R. Stanforth, J. Welbl, J. Rae, V. Maini, D. Yogatama, and P. Kohli, "Reducing sentiment bias in language models via counterfactual evaluation," *arXiv preprint arXiv:1911.03064*, 2019.

[110] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *arXiv preprint arXiv:1905.02175*, 2019.

[111] C. Zhu, Y. Cheng, Z. Gan, S. Sun, T. Goldstein, and J. Liu, "Freelb: Enhanced adversarial training for natural language understanding," *arXiv preprint arXiv:1909.11764*, 2019.

[112] X. Dong, H. Liu, R. Ji, L. Cao, Q. Ye, J. Liu, and Q. Tian, "Api-net: Robust generative classifier via a single discriminator," in *European Conference on Computer Vision*. Springer, 2020, pp. 379–394.

[113] S. Sharma, Y. Zhang, J. M. Ríos Aliaga, D. Bouneffouf, V. Muthusamy, and K. R. Varshney, "Data augmentation for discrimination prevention and bias disambiguation," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020, pp. 358–364.

[114] P. Czarnowska, Y. Vyas, and K. Shah, "Quantifying social biases in nlp: A generalization and empirical comparison of extrinsic fairness metrics," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1249–1267, 2021.

[115] Y. Deng, X. Zheng, T. Zhang, C. Chen, G. Lou, and M. Kim, "An analysis of adversarial attacks and defenses on autonomous driving models," in *2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2020, pp. 1–10.

[116] M. Gardner, Y. Artzi, V. Basmova, J. Berant, B. Bogin, S. Chen, P. Dasigi, D. Dua, Y. Elazar, A. Gottumukkala *et al.*, "Evaluating models' local decision boundaries via contrast sets," *arXiv preprint arXiv:2004.02709*, 2020.

[117] Y. Sharma and P.-Y. Chen, "Attacking the madry defense model with l_1-based adversarial examples," *arXiv preprint arXiv:1710.10733*, 2017.

[118] X. W. Li, S. J. Cho, and S. T. Kim, "High security and robust optical image encryption approach based on computer-generated integral imaging pickup and iterative back-projection techniques," *Optics and Lasers in Engineering*, vol. 55, pp. 162–182, 2014.

[119] K. Xu, Z. Shi, H. Zhang, Y. Wang, K.-W. Chang, M. Huang, B. Kailkhura, X. Lin, and C.-J. Hsieh, "Automatic perturbation analysis for scalable certified robustness and beyond," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[120] L. Derczynski, "Complementarity, f-score, and nlp evaluation," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016, pp. 261–266.

[121] A. Gautam and K. R. Jerripothula, "Sgg: Spinbot, grammarly and glove based fake news detection," in *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*. IEEE, 2020, pp. 174–182.

[122] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," *arXiv preprint arXiv:1806.03822*, 2018.

[123] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7472–7482.

[124] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," *arXiv preprint arXiv:1810.00069*, 2018.

[125] J. Li, X. Liu, W. Yin, M. Yang, L. Ma, and Y. Jin, "Empirical evaluation of multi-task learning in deep neural networks for natural language processing," *Neural Computing and Applications*, vol. 33, no. 9, pp. 4417–4428, 2021.

[126] L. Tu, G. Lalwani, S. Gella, and H. He, "An empirical study on robustness to spurious correlations using pre-trained language models," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 621–633, 2020.

[127] B. McCann, N. S. Keskar, C. Xiong, and R. Socher, "The natural language decathlon: Multitask learning as question answering," *arXiv preprint arXiv:1806.08730*, 2018.

[128] J. Bingel and A. Søgaard, "Identifying beneficial task relations for multi-task learning in deep neural networks," *arXiv preprint arXiv:1702.08303*, 2017.

[129] S. Subramanian, A. Trischler, Y. Bengio, and C. J. Pal, "Learning general purpose distributed sentence representations via large scale multi-task learning," *arXiv preprint arXiv:1804.00079*, 2018.

[130] M. Schmitt and H. Schütze, "Sherliic: A typed event-focused lexical inference benchmark for evaluating natural language inference," *arXiv preprint arXiv:1906.01393*, 2019.

[131] E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh, "Universal adversarial triggers for attacking and analyzing nlp," *arXiv preprint arXiv:1908.07125*, 2019.

[132] R. Zhao and K. Mao, "Fuzzy bag-of-words model for document representation," *IEEE transactions on fuzzy systems*, vol. 26, no. 2, pp. 794–804, 2017.

[133] D. Jatnika, M. A. Bijaksana, and A. A. Suryani, "Word2vec model analysis for semantic similarities in english words," *Procedia Computer Science*, vol. 157, pp. 160–167, 2019.

[134] M. Ibrahim, S. Gauch, O. Salman, and M. Alqahatani, "Enriching consumer health vocabulary using enhanced glove word embedding," *arXiv preprint arXiv:2004.00150*, 2020.

[135] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[136] Y. Peng, S. Yan, and Z. Lu, "Transfer learning in biomedical natural language processing: an evaluation of bert and elmo on ten benchmarking datasets," *arXiv preprint arXiv:1906.05474*, 2019.

[137] H. Gupta and M. Patel, "Study of extractive text summarizer using the elmo embedding," in *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE, 2020, pp. 829–834.

[138] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext. zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2016.

[139] B. Wang, A. Wang, F. Chen, Y. Wang, and C.-C. J. Kuo, "Evaluating word embedding models: Methods and experimental results," *APSIPA transactions on signal and information processing*, vol. 8, 2019.

[140] I. Santos, N. Nedjah, and L. de Macedo Mourelle, "Sentiment analysis using convolutional neural network with fasttext embeddings," in *2017 IEEE Latin American conference on computational intelligence (LA-CCI)*. IEEE, 2017, pp. 1–5.

[141] B. Athiwaratkun, A. G. Wilson, and A. Anandkumar, "Probabilistic fasttext for multi-sense word embeddings," *arXiv preprint arXiv:1806.02901*, 2018.

[142] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[143] D. Yogatama, C. d. M. d'Autume, J. Connor, T. Kocisky, M. Chrzanowski, L. Kong, A. Lazaridou, W. Ling, L. Yu, C. Dyer *et al.*, "Learning and evaluating general linguistic intelligence," *arXiv preprint arXiv:1901.11373*, 2019.

[144] B. Hutchinson, V. Prabhakaran, E. Denton, K. Webster, Y. Zhong, and S. Denuyl, "Social biases in nlp models as barriers for persons with disabilities," *arXiv preprint arXiv:2005.00813*, 2020.

[145] D. Shah, H. A. Schwartz, and D. Hovy, "Predictive biases in natural language processing models: A conceptual framework and overview," *arXiv preprint arXiv:1912.11078*, 2019.

[146] R. Le Bras, S. Swayamdipta, C. Bhagavatula, R. Zellers, M. Peters, A. Sabharwal, and Y. Choi, "Adversarial filters of dataset biases," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1078–1088.

[147] F. Möller, "Informed regularization aiding the identification of spurious correlations," *arXiv preprint arXiv:2004.05007*, 2021.

[148] Y. J. Choe, J. Ham, and K. Park, "An empirical study of invariant risk minimization," *arXiv preprint arXiv:2004.05007*, 2020.

[149] Y. Zhang, L. Pan, S. Tan, and M.-Y. Kan, "Causally estimating the sensitivity of neural nlp models to spurious features," *arXiv preprint arXiv:2110.07159*, 2021.