



Article

A Survey of Adversarial Attacks: An Open Issue for Deep Learning Sentiment Analysis Models

Monserrat Vázquez-Hernández ¹, Luis Alberto Morales-Rosales ^{2,*}, Ignacio Algreto-Badillo ^{1,*},
Sofía Isabel Fernández-Gregorio ³, Héctor Rodríguez-Rangel ⁴ and María-Luisa Córdoba-Tlaxcalteco ³

- ¹ Department of Computer Science, CONACYT-National Institute for Astrophysics, Optics and Electronics, Luis Enrique Erro #1, Sta María Tonanzintla, Puebla 72840, Puebla, Mexico; mvazquez@inaoe.mx
- ² Facultad de Ingeniería Civil, CONACYT-Universidad Michoacana de San Nicolás de Hidalgo, C. de Santiago Tapia 403, Centro, Morelia 58000, Michoacan, Mexico
- ³ Facultad de Estadística e Informática, Univesidad Veracruzana, Av. Xalapa, Obrero Campesina, Xalapa 91020, Veracruz, Mexico; zs21000291@estudiantes.uv.mx (S.I.F.-G.); cmarluit@gmail.com (M.-L.C.-T.)
- ⁴ Tecnológico Nacional de México, Instituto Tecnológico de Culiacán, Juan de Dios Batiz No. 310pte., Culiacán 80220, Sinaloa, Mexico; hector.rr@culiacan.tecnm.mx
- * Correspondence: lamorales@conacyt.mx (L.A.M.-R.); algretoabadillo@inaoe.mx (I.A.-B.)

Abstract: In recent years, the use of deep learning models for deploying sentiment analysis systems has become a widespread topic due to their processing capacity and superior results on large volumes of information. However, after several years' research, previous works have demonstrated that deep learning models are vulnerable to strategically modified inputs called *adversarial examples*. Adversarial examples are generated by performing perturbations on data input that are imperceptible to humans but that can fool deep learning models' understanding of the inputs and lead to false predictions being generated. In this work, we collect, select, summarize, discuss, and comprehensively analyze research works to generate textual adversarial examples. There are already a number of reviews in the existing literature concerning attacks on deep learning models for text applications; in contrast to previous works, however, we review works mainly oriented to sentiment analysis tasks. Further, we cover the related information concerning generation of adversarial examples to make this work self-contained. Finally, we draw on the reviewed literature to discuss adversarial example design in the context of sentiment analysis tasks.

Keywords: adversarial attacks; sentiment analysis; deep learning; vulnerabilities



Citation: Vázquez-Hernández, M.; Morales-Rosales, L.A.; Algreto-Badillo, I.; Fernández-Gregorio, S.I.; Rodríguez-Rangel, H.; Córdoba-Tlaxcalteco, M.-L. A Survey of Adversarial Attacks: An Open Issue for Deep Learning Sentiment Analysis Models. *Appl. Sci.* **2024**, *14*, 4614. <https://doi.org/10.3390/app14114614>

Academic Editors: Silvia García-Méndez, Enrique Costa-Montenegro and Francisco De Arriba-Pérez

Received: 14 April 2024
Revised: 21 May 2024
Accepted: 23 May 2024
Published: 27 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

User opinions allow researchers to identify the experience, either positive or negative, that actual consumers have had with a product, service, or topic of interest [1]. The opinions expressed by actual consumers play an influential role in the decision-making of new consumers and organizations. For example, potential consumers can take a look at the quality of a product or service through users' opinions and decide whether or not to select it. In the case of organizations, opinions allow them to determine necessary improvements to implement in their products or services to enhance their consumers' experience [2]. Lately, the volume of users' opinions has increased considerably, mainly due to the accessibility, anonymity, and free expression that digital media offer to users, becoming a comfortable space for users to express their opinions or feedback. Due to this large volume of information, tools are needed to facilitate simplified opinion analysis. In this context, sentiment analysis systems are an important tool for analyzing and providing summarized information concerning users' opinions to assist both users and organizations in the decision-making process.

Sentiment analysis (SA) is a Natural Language Processing (NLP) task that uses text analysis and machine-learning techniques to automatically extract and process users'

opinions [3]. The purpose of sentiment analysis systems is to provide summarized information and assist potential users and organizations in evaluating their selections; popular sentiment analysis applications include social media monitoring, customer support management, and analyzing customer feedback, among others. Sentiment analysis has been predominantly applied to text analysis (analysis of user opinions). However, SA can be applied to different modalities as well, such as visual, speech, and text [4]. In recent years, multimodal sentiment analysis models have been proposed that combine different modalities to obtain more accurate results [5]. In this paper, we focus on sentiment analysis considering text modality.

In recent years, deep learning (DL) models have become a popular research topic for better handling large volumes of information on sentiment analysis [6]. Deep learning models are potent algorithms that achieve excellent performance, allowing previous results to be outperformed in different areas. In sentiment analysis systems, the aim of implementing deep learning models is to improve the precision of results obtained with traditional machine learning algorithms and raise the confidence of users; however, this does not always turn out to be true. Figure 1 presents the general operation of a sentiment analysis model.

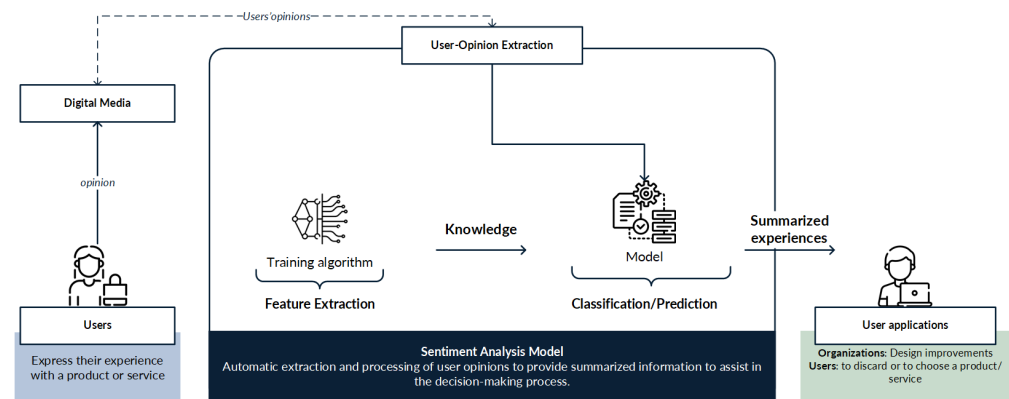


Figure 1. General operation of a deep learning-based sentiment analysis model.

After several years of research, it has been demonstrated both in theory and practice that DL models can be fooled with high probability by small modifications to the input data, causing incorrect results [7,8]. Research such as [7,8] has evaluated the robustness of deep learning models in face of small changes to the input data. Szegedy et al. [7] evaluated state-of-the-art deep neuronal network (DNN) models for image classification by making minor pixel-level modifications to inputs, identifying that DNN models can be fooled with a high probability by modified inputs causing classification errors even though people could not identify any change. The resulting images obtained after performing modifications called *adversarial examples*. Subsequently, this name has been applied to denote intentional modifications made to input data to fool models.

Based on the idea of adversarial examples, Jia and Liang [9] were the first to consider designing adversarial examples to fool DNN models for text applications (named textual deep neuronal networks). They generated short texts using words included in the training set of the neuronal network model. Then, the texts were added to the end of the original input data to generate the adversarial examples (also denominated adversarial texts), and it was observed that adversarial examples could also fool textual deep neuronal networks. Figure 2 shows an adversarial example designed in [9].

Led by [7], different approaches have been proposed that aim to: (1) design effective imperceptible modifications to confuse the DL models' understanding of inputs and cause incorrect results; (2) evaluate DL models against modified inputs; (3) propose defenses for potential modifications to input data; and (4) improve the robustness of DL models by adding new knowledge and capabilities via adversarial examples [10].

Since their introduction, adversarial examples have pointed to the limitations of deep learning models in correctly classifying modified inputs [11], which leads to the need to

understand their current vulnerabilities (and continuously explore new vulnerabilities) when facing adversarial examples. This can help to propose defenses that effectively guarantee confidence in deep learning models' results. Due to the importance of sentiment analysis in the decision-making process, we survey adversarial attacks on textual deep neuronal networks in this work, specifically focusing on sentiment analysis. In this work, we aim to summarize efforts that expose the vulnerabilities presented by deep neuronal sentiment analysis models when facing adversarial examples. Considering the importance of sentiment analysis and the impact of adversarial examples, there is a need for this kind of work to provide successive researchers with an overview of extant efforts.

Article: Super Bowl 50
Paragraph: "Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV."
Question: "What is the name of the quarterback who was 38 in Super Bowl XXXIII?"
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

Figure 2. Adversarial example from [9]. The assembly model originally obtains the correct answer, but is fooled by the insertion of an adversarial text (in blue).

Related surveys and differences with this survey. Reviews of adversarial examples attacking deep learning models for text applications already exist in the literature, such as [10,12–14]; however, to the best of our knowledge, no comprehensive review has collected and summarized the efforts in this research direction while specifically focusing on sentiment analysis. This study attempts to cover this gap via three central questions: (1) How do adversarial attack methods fool sentiment analysis models? (2) What proposals exist to defend against these attacks? and (3) What are the open challenges and directions of text adversarial attacks within sentiment analysis?

Paper selection methodology. We performed a literature review on three primary data sources related to computer science and data security: IEEE, ScienceDirect, and SpringerLink. First, keywords related to the central topic of this paper, namely, adversarial examples, were defined: (Adversarial, Examples, and Attack). Subsequently, keywords to delimit the research to sentiment analysis and text applications were added: (Text, Sentiment Analysis (SA), and Sentiment). Through the set of keywords defined above (Adversarial, Examples, Attack, Text, SA, Sentiment), the following query string was integrated: (Adversarial Attacks in Sentiment Analysis, Adversarial Attacks in Text). Considering that the research topic for text applications has emerged only recently, we limited the search to papers published from 2017 to January of 2024. The executed query string was as follows: ("adversarial attack" or "adversarial texts") and ("text" or "sentiment" or "sentiment analysis").

The inclusion and exclusion criteria for filtering and selecting papers were formally defined as follows:

- The inclusion criteria considered the following elements:
 1. The paper's contributions focus on the design of adversarial attack or defense mechanisms, particularly for the sentiment analysis task.
 2. Priority was given to papers published in the most recognized conferences on Natural Language Processing and Artificial Intelligence (ACL: Annual Meeting of the Association for Computational Linguistics; COLING: International Conference on Computational Linguistics; EMNLP: Empirical Methods in Natural Language Processing; IJCAI: International Joint Conference on Artificial Intelligence), although this was not a limitation.

- On the other hand, the following exclusion criteria were defined:
 1. Papers unrelated to text applications.
 2. The content and contributions of the papers were not related to the research subject, i.e., adversarial attacks and their application in sentiment analysis.
 3. Papers unrelated to sentiment analysis with deep neural network approaches.
 4. The method presented in the paper was largely derived from other work.

Through the search engines the selected information sources, the query string was executed for all metadata in the advanced search section. Figure 3 depicts the process of extracting and filtering works proposing adversarial attack designs for the study task. Figure 3a illustrates the amount of articles retrieved in the data extraction stage by executing the query string. We obtained a total of 1040 articles related to adversarial attacks for text applications, including articles for sentiment analysis. First, the retrieved works were manually filtered by checking the year of publication, title, and abstract to exclude those that did not align with the research objectives of this work. As a result, the number of retrieved works was reduced to 51 (refer to Figure 3b). As a final step, the inclusion and exclusion criteria were applied, resulting in 33 papers being selected for the initial review.

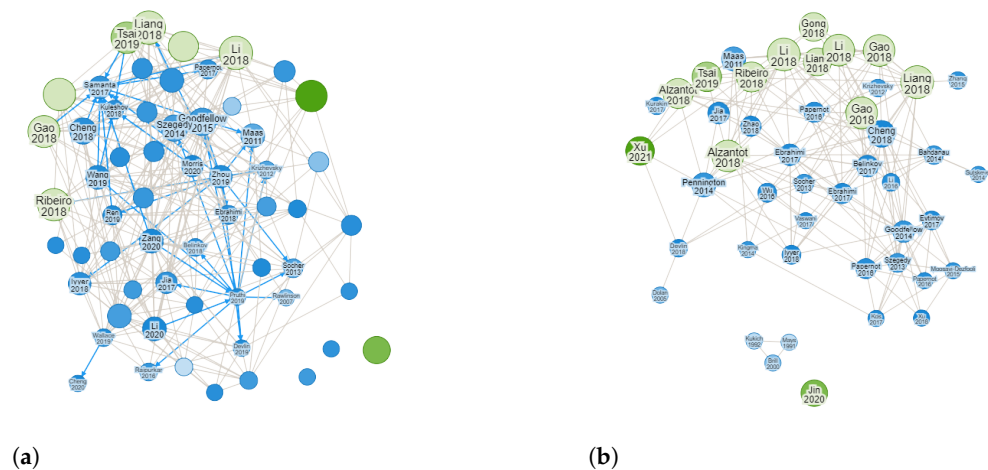


Figure 3. Paper selection methodology. After applying the query string, a total of 1040 papers related to adversarial attacks for text applications and sentiment analysis were retrieved in the data extraction stage. These papers were filtered by reviewing the year, title, and abstract, then applying the inclusion and exclusion criteria after which the total number of papers for initial review was reduced to 33. (a) Papers retrieved during the data extraction stage. A total of 1040 works were obtained for text applications, including works addressing sentiment analysis attack. The green circles highlight the primary works from which others were derived according to our query string. (b) The papers obtained during data extraction were filtered by checking the year, title, and abstract to validate the period and determine whether they corresponded to the topics of adversarial attacks and sentiment analysis. A total of 51 papers remained after filtering. Application of the inclusion and exclusion criteria further reduced the total number of papers for initial review to 33.

Contributions of this survey. The objective of conducting this survey is to provide a comprehensive review of research efforts on adversarial example generation, specifically for deep neural networks used in sentiment analysis. In order to comprehend the importance of sentiment analysis in decision-making and knowing the scope that adversarial examples have on the results of deep learning models, it is necessary to understand the behavior of adversarial attacks with the aim of proposing defenses against them. We pursue this survey to serve researchers and practitioners interested in designing, attacking, and defending sentiment analysis deep neuronal networks. Although this work includes essential information concerning preliminary knowledge related to adversarial attacks, we expect that readers will have basic knowledge of sentiment analysis and deep neural network

architectures, which are not the focus of this article. To summarize, the main contributions of this survey are:

- We collect, select, review, summarize, and discuss adversarial attacks, especially those suited to sentiment analysis models, with the objective of providing an information framework for current adversarial attacks methods that expose vulnerabilities of sentiment analysis models. This is the first survey to mainly focus on adversarial attacks for sentiment analysis. The aim is to assist researchers in addressing new vulnerabilities and developing defense methods to mitigate the negative consequences that can result from incorrect results produced by sentiment analysis models.
- We provide essential information concerning preliminary knowledge related to adversarial attacks, which allows this work to be self-contained.
- We draw on the reviewed literature to discuss open issues in adversarial example design for sentiment analysis deep neuronal network models, and identify possible new research directions concerning defenses for sentiment analysis models based on the issues and challenges identified during this review.

The rest of this document is organized as follows. Section 2 presents concepts related to the design of adversarial attacks for deep learning models in the sentiment analysis task, which allows for a better understanding of the primary state-of-the-art works regarding adversarial attacks presented in Section 3. The main works proposed for the sentiment analysis task involving defense against adversarial examples are presented in Section 4. Finally, a discussion of the current challenges in adversarial example design are described in Sections 5 and 6 presents the conclusions of this work.

2. Adversarial Examples and Deep Learning Models

Before introducing the reviewed works concerning adversarial examples and sentiment analysis models, we introduce some preliminary knowledge related to adversarial attacks and their application with regard to textual deep learning models, with the objective of providing an overview of how adversarial examples work in these types of applications.

2.1. Definitions

The principal definitions to consider when adversarial examples are studied are the following:

- **Deep Neuronal Network (DNN).** A deep learning model is a set of machine learning algorithms that attempts to model high-level abstractions using neural network architectures that support multiple and iterative nonlinear transformations of data expressed in matrix or tensor form [15].
- **Perturbation.** The perturbations η are small modifications intentionally made to input data in order to confuse the deep learning models.
- **Adversarial Example.** An adversarial example is a modified input created via a perturbation η of the input x of a deep learning model. The perturbation η is the minimal worst-case modification to input data that succeeds in confusing the model in its understanding and, as a consequence, in its classification. A robust model should continue to classify the correct class y to x' , while a victim model will have a high probability of incorrectly classifying x' . Equation (1) presents the formalization of x' :

$$\begin{aligned} f(x) &= y, \quad x \in X \\ x' &= x + \eta, \quad f(x') \neq y \\ \text{or } f(x') &= y', \quad y' \neq y \end{aligned} \quad (1)$$

where n is the worst-case perturbation. The goal of the adversarial examples is to cause the label to deviate to an incorrect label $f(x') \neq y$ or to a specific label $f(x') = y'$. Modifications to create adversarial examples should be as small as possible while being capable of fooling DL models without changing human perception.

2.2. Adversarial Examples for Text Applications

In recent years, the design of adversarial examples has attracted research interest with the aim of identifying the vulnerabilities of DNN models against modified inputs and thereby proposing defense mechanisms that guarantee the safety of the results. The original idea of adversarial examples was explained by Szegedy et al. [7] when evaluating the robustness of image classification DNN models by making minor pixel-level modifications. Nevertheless, when evaluating a textual DNN model, it is impossible to apply the same methods from evaluating image DNN due to three main differences:

1. **Input space.** Image inputs are continuous, while text data are symbolic, i.e., discrete. Thus, it is hard to define perturbations in texts that maintain the semantic and syntactical properties of the inputs.
2. **Perceptibility.** While small changes in image pixels are usually difficult to perceive, small changes in text, e.g., changing characters or words, can be much more easily perceived.
3. **Semantics.** In the case of images, small changes usually do not change the semantics of the image, while perturbation of text can easily change the semantics of words or sentences.

Considering these principal differences, works involving adversarial examples for textual DNNs have proposed novel methods using different techniques to carefully preserve the semantic inputs and the imperceptibility of modifications. Jia and Liang [9] were the first to consider designing adversarial examples for text applications with DNN models. In their work, they experimented by inserting small text fragments into the input data. These fragments were created synthetically based on observed terms in the training set (refer to Figure 2). Following Jia and Liang [9], different adversarial works oriented to text-based applications have shown that performing modifications at the character, term, or sentence level by inserting, deleting, substituting, or exchanging characters or terms can cause models to produce incorrect results [16–19]. When generating adversarial examples, the modifications should be as small as possible while remaining capable of fooling the model. For text-based tasks, the modifications should not make drastic changes to the text's semantics or syntax, and should maintain the readability of the input message.

2.3. Taxonomy of Adversarial Attacks

Generating adversarial examples is motivated by one of two objectives: (i) attacking by confusing the deep learning model's understanding of inputs to cause incorrect results; or (ii) defending by improving the robustness of the model through the addition of new knowledge and capabilities [10]. Adversarial example attacks attempt to identify and exploit model vulnerabilities when facing modified inputs, allowing effective defenses to subsequently be proposed to cover these vulnerabilities.

To identify the best criteria for designing adversarial examples, it is advisable to develop, test, and analyze different modifications to determine which will most effectively fool the target model while preserving the semantics and syntax of the text input. Zhang et al. [10] proposed a taxonomy of adversarial attacks on text application models based on the criteria established by Yuan et al. [20] for the design of a threat model or attack. According to [10,20], adversarial attacks can be classified as follows: (i) *model access*; this refers to the operational knowledge of the model under attack when the attack is performed; (ii) *semantic application*; while this can refer to different NLP applications, such as machine translation, machine comprehension, and speech recognition, among others, in this survey we focus on discussing the application of adversarial attacks to sentiment analysis; (iii) *target*; this refers to the objective of the attack, which may be to pursue an incorrect prediction or to direct the results to be specific to a class; (iv) *granularity*, which considers the level of text granularity at which the models are attacked; and (v) *attacked DNNs*, which indicates the DNN architecture used by the attacked model. We discuss the main models used in sentiment analysis in Section 2.6. In the following sections, we describe the different groups to which adversarial attacks can belong.

2.3.1. Model Access

According to the attacker's knowledge about the model to be fooled (or victim model), three types of attacks can be carried out: *black box*, *white box*, and *grey box*. A victim model can suffer attacks under different levels of knowledge at the same time, or the attacks can be generated independently.

- **Black box.** Black box attacks are applied when the architectures, parameters, activation, or loss functions are not accessible by the attacker. In this case, adversarial examples are generated by accessing the test dataset or querying the target model by making requests until a modification that allows for a change in the output is found.
- **White box.** Unlike black box attacks, white box attacks rely on knowledge of the complete details of the target model to be fooled.
- **Grey box.** Gray-box attacks occupy a middle ground between black box and white box attacks.

2.3.2. Target

The adversarial examples generated from an attack can be designed to change the model's output to make it incorrect $f(x') \neq y$, that is, an untargeted attack, or to change the output to a specific result $f(x') = y'$, that is, a targeted attack (as is indicated in Equation (1)). Targeted attacks are more strict compared to untargeted attacks, as they both change the prediction output and impose constraints on the output to produce a specific prediction [21].

2.3.3. Granularity

Different works oriented to text-based tasks have shown that it is possible to cause incorrect results by performing modifications at the character, term, or sentence level by inserting, deleting, substituting, or exchanging characters or terms [16–19]. To generate adversarial examples, the modifications must be as small as possible while being capable of fooling models. Furthermore, for text-based tasks, the modifications should not make drastic changes to the semantics and syntax, and should maintain the readability of the input message. In text inputs, modifications to generate adversarial examples can be performed at a different levels of detail, including character, term, sentence, and multilevel.

- **Character.** Modifications at the character level consist of modifying one or more characters within a term in an attempt to preserve its structure. Possible modifications include insertion, deletion, swapping, and replacing [22].
- **Term.** Modifications at the term-level consist of modifying a term (simple or n-gram) within a text while attempting to preserve the semantics and syntax. Modifications at this level include insertion, deletion, swapping, and replacing a term with a synonym or antonym [23].
- **Sentence.** Modifications at the sentence level mainly involve reordering terms by paraphrasing the sentence while maintaining the message's meaning [24]. Advanced methods aim to insert fragments of the text created based on the terms in the dataset.
- **Multilevel.** Multilevel modifications combine changes at the character/term/phrase levels with the aim of identifying the optimal change to be performed [25,26].

2.4. Strategies for Performing Modifications

Modifications can be performed through different strategies, such as modifying certain terms or a complete text in an original input according to the level of granularity. Text-based strategies for input modification include the following:

- **Concatenation.** Concatenation consists of adding a sentence called a distract text at the end of a text to confuse the model without changing the semantics of the message [9]. In this strategy, a distract text is added to the original inputs, then requests are made to the target model until the output is modified. When the target

model is successfully confused and incorrect results are generated, the distract text is identified and added to the original input to create the adversarial example (Figure 4).

- **Editing.** The editing strategy (Figure 5) performs modifications to input data in two ways: (i) *synthetic*, in which a change in the order of the characters is made, which can be through *swapping*, *middle random* (random characters are exchanged except for the first and the last one), or *fully random* (all the characters are randomly rearranged and the keyboard type is changed); or (ii) *natural*, in which spelling errors in the original data are exploited. Advanced applications carry out modifications such as *random swap* by making an exchange of neighboring terms, *stop-word dropout* by randomly removing empty words, *paraphrasing* by substituting terms with paraphrased text, and *grammar errors* by modifying the conjugation of a verb, as well as *add negation* and *antonym* strategies.
- **Paraphrasing.** This strategy carefully produces a paraphrase of the original entry (Figure 6).
- **GAN-based strategies.** The purpose of adopting a generative adversarial network (GAN) architecture is to make adversarial examples seem more natural [10]. Generally, these attacks consist of two components: a GAN used to design adversarial examples, and an inverter that maps the input x to its representation within the latent space.
- **Substitution.** This strategy involves substituting terms with related terms, e.g., substitution by synonym.
- **Hybrid strategies.** The different strategies mentioned above can be hybridized to generate adversarial examples.

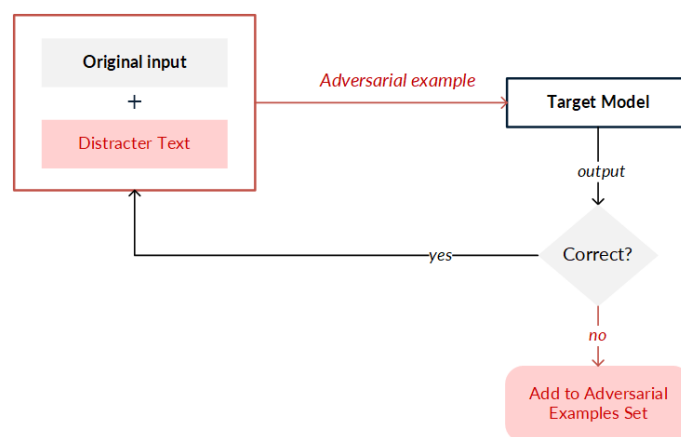


Figure 4. General principle for creating adversarial examples (or adversarial texts) by implementing concatenation.

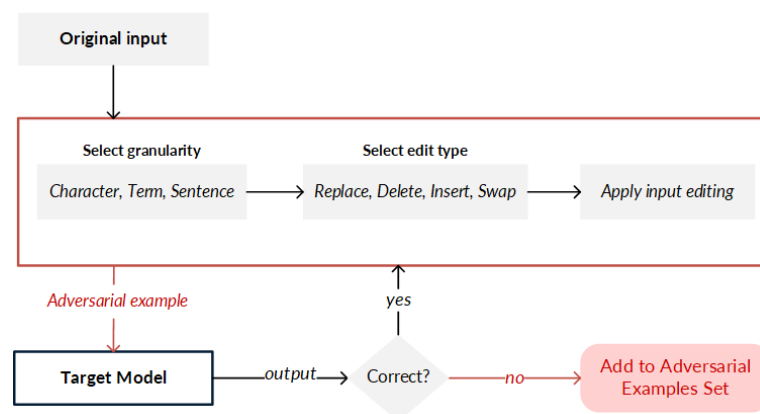


Figure 5. General principle for creating adversarial examples (or adversarial texts) by implementing editing. Modifications are made to sentences, terms, or characters by substitution, deletion, addition, or swapping.

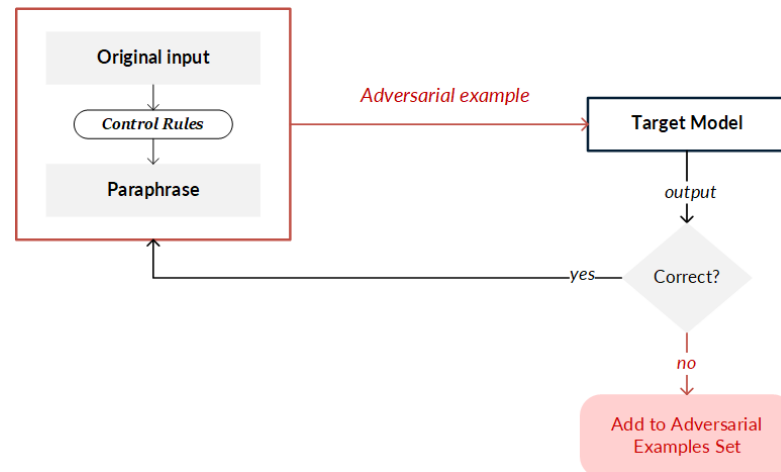


Figure 6. General principle for creating adversarial examples (or adversarial texts) by implementing paraphrases. Modifications are made carefully to ensure that the syntax and grammar of the input are not changed.

2.5. Measurement and Evaluation Metrics

During the development of an adversarial attack, it is necessary to measure and control the perturbations or modifications in order to keep their size to a minimum and ensure that they are imperceptible. Afterwards, the effectiveness of the attack in terms of the negative impact on model results and number of successfully misclassified modified inputs needs to be evaluated. The following measures are usually used to control modifications and measure the effectiveness of an attack.

2.5.1. Modification Control

After modifying the input data, it is necessary to measure the size of the modifications in order to ensure that they are unnoticeable. Usually, the size of the modifications is measured based on the distance between the original data (or *clean data*) x and adversarial example x' . In the case of text data, the distance between x and x' must be measured. Correct grammar, syntax, and semantic preservation must be considered as well.

- **Grammar and syntax measurement.** Ensuring correct grammar and syntax is necessary to ensure that adversarial examples are undetectable. Strategies such as perplexity measure, paraphrase control, and grammar and syntax checkers have been proposed to measure grammar and syntax.
- **Semantic-preserving measurement.** The semantic similarity/distance measurement is performed on word vectors using measures of distances (such as the Euclidean distance) and similarity (such as the cosine similarity).
 - The Euclidean distance is the distance between two vectors in Euclidean space.

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots (p_n - q_n)^2} \quad (2)$$

- The cosine similarity computes the cosine value of the angle between the two vectors.

$$\cos(p, q) = \frac{\sum_{i=1}^n p_i \times q_i}{\sqrt{\sum_{i=1}^n (p_i)^2} \times \sqrt{\sum_{i=1}^n (q_i)^2}} \quad (3)$$

- **Edit-based measurement.** Measuring the number of edits (modifications) quantifies the minimum changes from one text to the next. Different definitions of editing distances use different operations:

- Word Mover's Distance (WMD). The WMD measures the changes in the space of word embeddings. It measures the minimum distance from the word embeddings of an adversarial text to approach the *word embeddings* of an original text.
- Levenshtein distance. The Levenshtein distance is a string metric for measuring the difference between two sequences, i.e., the minimum number of single-character edits.
- Perturbation ratio. The ratio of perturbed words in the sentence to the total number of words in the sentence.
- Jaccard similarity coefficient. Used to measure the similarity of finite sets using the intersection and union of the sets:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (4)$$

in two texts, A and B are the original and modified inputs, respectively, $|A \cap B|$ indicates the number of terms in both documents, and $|A \cup B|$ is the number of unique terms in both documents.

2.5.2. Attack Evaluation

During the development of an adversarial attack, it is necessary to measure and control the perturbations or modifications to keep their size to a minimum and ensure that they are imperceptible. Afterwards, the effectiveness of the attack must be evaluated.

- **Success rate.** The success rate is the most direct and effective evaluation criteria [27]. The attack success rate indicates the percentage of successful adversarial examples and the percentage of unsuccessfully attacked inputs. This measure provides insight into the susceptibility of a model to the designed adversarial examples.
- **Model Robustness.** Adversarial attacks are designed to affect the performance of models concerning the correct classifications. The robustness of DL models is related to the classification accuracy (*before-attack accuracy*, or BA) and how it is affected by adversarial examples (*after-attack accuracy*, or AA).

2.6. Deep Neural Networks

Deep learning models allow the characteristics of the input data to be learned at various abstraction layers, which in turn allows systems to learn the most complex functions to identify characteristics of the study domain. Feedforward, Convolutional Neural Network (CNN), and Recurrent/Recursive Neural Network (RNN) models and their variants have been the most widely implemented models for sentiment analysis tasks due to their natural ability to handle sequences and understand the relations between different elements. In particular, Long Short-Term Memory Network (LSTM) and Convolutional Neural Network (CNN) models are able to learn about sequences both locally and in the long term, preserving the most important and complex features to help the model understand the complete relationships. Recently, two major advances in deep learning have been popularized to address text-related tasks: (1) *sequence-to-sequence learning* [28] and (2) *attention mechanisms* [29].

- **Recurrent Neural Networks.** RNN models can handle input sequences with variable lengths. RNNs create and process arbitrary memory sequences of input patterns. Unlike traditional methods for automatic sequence synthesis, RNN models can process sequential and parallel information naturally and efficiently [30].
- **Long Short-Term Memory Networks.** LSTM models are a particular type of RNN composed of units of the same type. Conventional RNN models can encounter problems during training, as the gradients tend to grow enormously or fade over time due to their dependence on present and past errors. The accumulation of errors can cause difficulties when memorizing dependencies in long texts. LSTMs tackle these problems by incorporating decisions about which information will be stored and which will be discarded [31].

- **Convolutional Neural Networks.** CNN models consist of multiple layers of convolution filters of one or more dimensions. After each layer, a function is added to perform nonlinear causal mapping. At the beginning of the CNN, the feature extraction phase is composed of convolutional and downsampling neurons; as more data are processed, its dimensionality decreases, with the neurons in distant layers being much less sensitive to data perturbations while being activated by increasingly complex features (<https://www.juanbarrios.com/redes-neurales-convolucionales/>) (accessed on 12 April 2024).
At the end of the network, perceptron neurons perform the final classification on the extracted features.

3. Adversarial Attacks on Sentiment Analysis

Sentiment analysis systems are an important tool that provides summarized information on users' opinions to assist potential users and organizations in the decision-making process. Sentiment analysis systems attempt to determine the user experience with respect to a product or service based on the positive or negative connotations of the words used by users to express their opinions [3]. In recent years, the use of deep learning models for deploying sentiment analysis systems has become a widespread topic due to their good processing capacity and superior results achieved on large volumes of information. However, as described above, deep learning models are vulnerable to modified inputs, making sentiment analysis models vulnerable to adversarial attacks. This issue leads us to study how adversarial attacks operate in order to propose defenses that minimize the negative impact that adversarial examples can have on the results of sentiment analysis models, which could potentially affect decision-making.

According to the characteristics of the reviewed works, and considering the discussed criteria in [20] for threat model design, we propose a new taxonomy specifically focused on adversarial attack methods for sentiment analysis models. Figure 7 presents the proposed taxonomy. In this taxonomy, six criteria are used to categorize the attack methods: (i) *model access* refers to the knowledge of the attacked model when the attack is performed; (ii) *analysis level* refers to the sentiment analysis level approached by the model under attack; (iii) *granularity* refers to the level of granularity at which the modifications are made; (iv) *element selection* refers to the way the methods select the element to be modified; (v) *strategy* indicates the strategy by which the modification is performed; and (vi) *DNN model* specifies the type of DNN being attacked. The main DNN architectures used for text applications are discussed in Section 2.6.

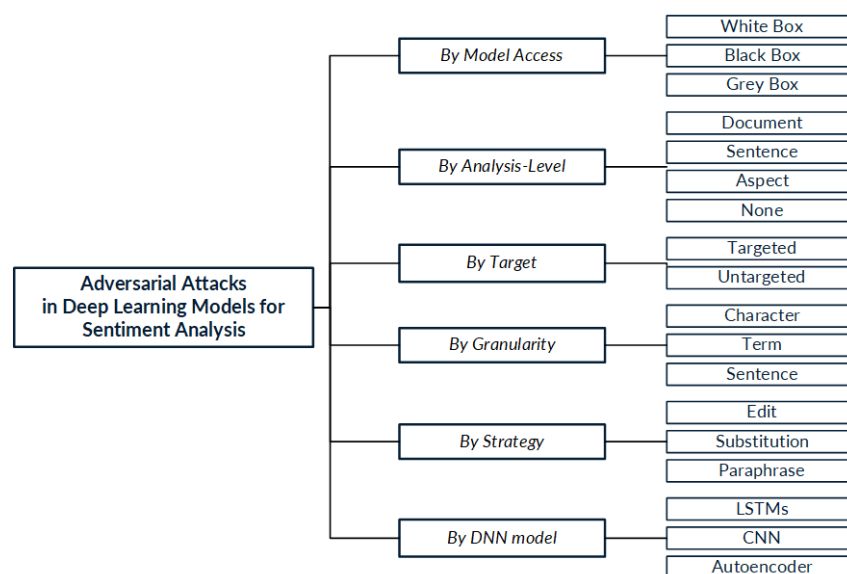


Figure 7. Taxonomy of adversarial attacks on deep learning models used for sentiment analysis.

3.1. Adversarial Attacks

The principal objective of sentiment analysis models is to obtain an effective set of terms that can uniquely identify different sentiments (positive, negative, or neutral), contributing to the classification of an opinion. Previous authors have referred to these terms as *valuable words* on account of their a crucial role in the final classification [32,33]. According to [34], to identify the most important terms in a text, it is necessary to consider two essential questions:

1. *Why is a text classified as positive or negative in a DNN model of sentiment analysis?*
2. *Do all words in an opinion contribute to the classification of an input to the same degree?*

The existing adversarial attacks seek to precisely determine those opinion terms that contribute to the correct input classification, then use them to perform modifications and create adversarial examples. However, when developing adversarial attacks, the ability to identify the most important terms in a text which impact the final classification of an input by a model is affected by the attacker's level of knowledge about the victim model.

Figure 8 illustrates that adversarial attacks on a sentiment analysis model may depend on the victim model's level of knowledge. In white box attacks, attackers have full knowledge of the target model to be attacked, including its data, architecture, and parameters, allowing adversarial examples to be created by using this knowledge directly. Typically, white box attacks modify the training data to make the model incorrectly learn features, producing incorrect results. On the other hand, in black box attacks the attackers do not know the target model's data or structure; thus, attackers can only consult the query output by the victim model to generate adversarial examples. In black box attacks, adversarial examples are generated by applying heuristics in a local model that represents the target model, which is trained until modifications that change the results are found. Usually, black box attacks collect representative data to generate adversarial examples; later, the adversarial examples generated by the substitute model are introduced to the victim model to cause incorrect results. Finally, in gray box attacks, the knowledge of the attackers is limited to the training data and general structure of the model to be attacked, and modifications are made using self-analysis methods to identify the most important parts of an input.

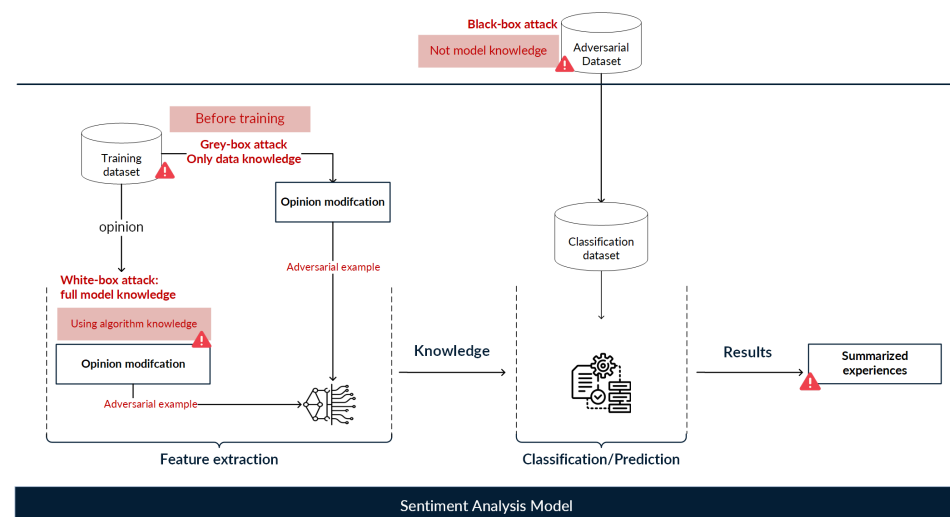


Figure 8. General overview of adversarial attacks on sentiment analysis models. Adversarial attacks can be classified as white box, black box, or gray box attacks depending on the attackers' level of knowledge about the data, architecture, and parameters of the victim model.

Table 1 summarizes the adversarial attacks reviewed in this paper. According to the proposed taxonomy (refer to Figure 7), we indicate each reviewed attack's principal criteria: model access, sentiment analysis level attacked, granularity, and strategy applied to perform input modifications. Additionally, we include the type of modification control

used to perform the modifications. In the following sections, we present the reviewed methods separately based on the knowledge of the victim model. First, in Section 3.2, we discuss white box attacks, then, in Section 3.3, we introduce black box and gray box attacks.

Table 1. Reviewed adversarial attacks on sentiment analysis models.

Work	Sentiment Analysis Level	Model Knowledge	Modification Granularity	Select Elements to Be Modified via	Modification Strategy	Modification Control
[25]	-	White-box	Character/Sentence	Word Importance on final prediction	Substitution	-
[16]	-		Term	-	Edit. Reconstruct the sentences via nearest semantic neighborby term	Word Mover's Distance
[18]	-		Term/Character	Word Importance on final prediction	Edit. Synthetic	Edit distance, Jaccard similarity, Euclidean distance and Semantic similarity
[17]	-	Black-box	Term	-	Substitution	Semantic Similarity
[19]	-		Term	-	Substitution	Euclidean Distance
[35]	-		Sentence	-	Edit. Paraphrases	Semantic similarity
[36]	-		Term/Character	Word Importance on final prediction	Edit. Eliminating one token at a time	-
[37]	-		Term	-	Substitution	Semantic Similarity
[38]	Aspect	Grey-box	Term	Word Importance on final prediction	Substitution	Semantic Similarity
[39]	-		Term	Word Importance on final prediction	Substitution	Semantic Similarity

3.2. White Box Adversarial Attacks

In white box attacks, the attack requires full access to the model's information, including its architecture, parameters, loss functions, activation functions, and input and output data. White box attacks typically approximate the worst-case attack scenario.

- Liang et al. [25] presented *TextFool*, a targeted attack that uses the FGSM (Fast Gradient Sign Method) concept to approximate the contribution of elements in a text in order to identify those that significantly impact the classification of the input text. Instead of using the sign of the cost gradient in FGSM, this work considers the magnitude. Specifically, the authors computed the cost gradient $\Delta_x J(f, x, c')$ of each instance in the training set by employing backpropagation, where f is the model function, x is the original data sample, and c' is the target text class. After that, *hot characters* were identified, which are the characters with the highest grand dimensions. The entries containing sufficient *hot characters* were denoted as *Hot Training Phrases* (HTP). Using HTP, the adversarial examples were created by implementing three types of modifications: (i) *insertion*, in which HTPs of the target class are inserted into c' nearby significant entries of the original class c ; (ii) *modification*, in which some characters in the HTPs are replaced and *Hot Sample Phrases* (HSPs) are created; and (iii) *deletion*, in which unnecessary adjectives or adverbs in the HSPs are eliminated. These three strategies and their combinations were evaluated using a CNN architecture.
- Gong et al. [16] proposed an untargeted attack to perturb text inputs in the word embedding space of a CNN model. The proposed method addresses two main problems when performing text modifications, namely, the input space and the quality of adversarial examples. The main difficulties in generating adversarial texts are: (i) the text input's discrete space, which makes it difficult to accumulate small noise samples; and (ii) measuring the quality of adversarial texts. To deal with the input space issue, the authors perturbed the text inputs in the space of *embeddings* against a CNN model. Specifically, they applied FGSM and DeepFool. However, applying methods from computer vision would generate meaningless adversarial texts. To address this, the authors rounded the adversarial examples to the nearest meaningful word vectors, using the word mover's distance measure (WMD) as the distance measurement. Their evaluation on sentiment analysis and text classification datasets showed WMD to be a qualified metric for controlling perturbations.
- Li et al. [18] presented an untargeted attack method called TextBugger, which provides a perturbation constraint by using similarity measures such as edit distance, Jaccard similarity coefficient, Euclidean distance, and cosine similarity. To generate

adversarial examples, the proposed method first finds important words by computing the Jacobian matrix for the given input text to find the confidence value of each word, then finds the important words that have a significant impact on the classifier's outputs. By finding the most significant words, adversarial examples are generated by five kinds of modifications at the character or word level: (i) inserting a space into the word; (ii) deleting a random character of the word (except for the first and the last character); (iii) randomly swapping two adjacent letters; (iv) Substitute-C (Sub-C), in which characters are replaced with visually similar characters; and (v) Substitute-W (Sub-W), in which a word is replaced with its top_k nearest neighbors. For character-level perturbations, the authors deliberately misspelled important words to convert these important words to "unknown" words. For word-level perturbations, they used a semantic-preserving technique, i.e., replacing the word with its top_k nearest neighbors in a context-aware word vector space. In addition, the authors presented the application of their method under a black box setting while making word-level modifications. The process of generating adversarial examples contained three steps: (i) finding important sentences that contribute most to the final prediction results; (ii) using a scoring function to determine the importance of each word to the classification result and ranking the words based on their scores; and (iii) making modifications to selected words, similar to the white box approach. The attack methods were evaluated on CNN and LSTM architectures.

- Tsai et al. [17] presented an untargeted method called *Greedy Search*. Given a text input, this method considers the k nearest neighbors of each word in the space of *embeddings*. The greedy approach forms adversarial examples by replacing the original word w with each candidate w' among the k neighbors and determining whether the sigmoid value of the adversary x' is less than the value σ , which indicates whether replacing w with w' can contribute to change the prediction result. In addition, a more sophisticated approach called *Global Search* was proposed, in which simple modifications are performed by adding spelling error noise. *Global Search* computes small perturbations δ to the original word embeddings. To learn the perturbations δ , an objective function $J(\delta)$ is defined to maximize the difference between the sigmoid values of the original input x and the perturbed input $x + \delta$:

$$J(\delta) = (f_{\text{sigmoid}}(E_x) - f_{\text{sigmoid}}(E_x + \delta))^2 + \lambda_1 \cdot \|\delta\|_2 + \lambda_2 \cdot \|(E_x - (E_x + \delta))\|_2 \quad (5)$$

where λ_1 penalizes large perturbations and λ_2 penalizes large distances between the original and perturbed word embeddings. The two regularization terms are added to help maintain the semantics of the chosen words. The perturbed embedding $E'_x = E_x + \delta$ usually does not have an actual word that it can be mapped back to; thus, the attack algorithm finds the candidate words w in the embedding space that are the closest to the perturbed word embedding. After computing the perturbed embedding and recording the candidate words, the algorithm checks whether the current perturbed embedding changes the prediction result. The algorithm continues to compute new samples E' until it fools the model. Both attacks were evaluated on a CNN using the IMDB movie reviews dataset.

- Alzantot et al. [19] used a population-based optimization algorithm to generate semantically and syntactically similar adversarial examples that fool sentiment analysis and textual entailment models. The proposed attack algorithm exploits population-based gradient-free optimization via genetic algorithms. Given an input sentence x , the algorithm randomly selects a word w in x and then selects a suitable replacement word that has similar semantic meaning, fits within the surrounding context, and increases the target label prediction score. To select the replacement word, for each w in x , the nearest N neighbors in the space of *embeddings* are determined by computing the Euclidean distance and selecting words greater than δ that are synonyms of the

term to be replaced. Then, using the *Google one billion words language model*, the identified synonyms that are less frequent in the context of the text are discarded, keeping only the top K words. Finally, from the K remaining terms, those that contribute most to the classification when substituting the original term are selected. The attack algorithm was evaluated on an LSTM architecture and a textual entailment model trained on the Stanford Natural Language Inference (SNLI) corpus using the IMDB movie reviews dataset.

3.3. Black Box and Gray Box Adversarial Attacks

Black box attacks do not require the details of the neural networks, although they can access the input and output. This type of attack often relies on heuristics to generate adversarial examples, and is more practical in real-world applications.

- Ribeiro et al. [35] exploited the paraphrasing strategy to create semantically equivalent adversarial examples (SEA). They generated paraphrases of an input text x and observed the model predictions from $f(x')$ until the original prediction was modified. To control the generation of paraphrases, the authors defined an indicator function $SemEq(x, x')$ that is 1 if x is semantically equivalent to x' and 0 otherwise. They defined a semantically equivalent adversary (SEA) as a semantically equivalent instance that changes the model prediction:

$$SEA(x, x') = 1[SemEq(x, x') \wedge f(x) \neq f(x')]. \quad (6)$$

To generate adversarial samples, a set of paraphrases Πx around x was generated via beam search to obtain predictions on Πx using the victim model until an adversary was found or until $S(x, x_0) < \tau$. Additionally, a semantic equivalent rule-based method was proposed to generalize adversarial examples to understand and correct the failures when generating paraphrases. The rule takes the form $r = (a \rightarrow c)$, where the first instance of the antecedent a is replaced by the consequent c for every instance x that includes a . The authors generated SEAs and proposed rules for every $x \in X$.

- Gao et al. [36] proposed the DeepWordBug method for generating small text perturbations in a black box environment. In this method, the *Replace-1 Score (R1S)*, *Temporal Head Score (THS)*, *Temporal Tail Score (TTS)*, and *Combined Score (CS)* punctuation strategies are used to identify key terms that cause the classifier to make an incorrect prediction when modified. Character-level transformations can then be performed on the most relevant terms to minimize the edit distance of the perturbation from the original input.
- Jin et al. [37] presented the TextFooler method, which uses the fundamental NLP tasks of text classification and textual entailment to generate adversarial examples. When provided with a sentence, the proposed approach selects the words that most significantly influence the final prediction results through a selection mechanism that measures the influence of a word $w_i \in X$ on the classification result $F(X) = Y$. Each word is removed from the sentence one at a time in order to calculate the word importance, then the prediction score for label Y is measured. The importance score I_{w_i} is then calculated as the prediction change before and after deleting the word w_i . When the words with a high importance score have been obtained, they are replaced by the closest synonyms according to the cosine similarity between w_i and every other word. The proposed attack was evaluated on CNN and LSTM architectures, including pretrained BERT.
- Xu et al. [39] presented a gray box adversarial attack and defense framework which consists of a generator ζ (updated) and two copies of a pretrained target classifier: a static classifier C and an updated/augmented classifier C^* . During the training phase, the output of ζ is directly fed to C and C^* to form a joint architecture. Post-training, the generator ζ is used independently to generate adversarial examples (adversarial attack), while the augmented classifier C^* is an improved classifier with increased robustness (adversarial defense). The training phase is divided into attack and defense,

where the former updates only the generator ζ and learns to introduce slight perturbations to the input by maximizing the objective function of the target model and the latter updates C^* and ζ by feeding both original examples and adversarial examples generated by ζ . Here, the adversarial examples are assumed to share the same label as their original examples. The defending steps consist of training an improved classifier with data augmented by adversarial examples. The generator ζ is implemented as an autoencoder or a paraphrase generator. The proposed attacks were evaluated on CNN and Bi-LSTM architectures and a C-BERT model obtained by fine-tuning the BERT-Base model.

4. Defense against Adversarial Attacks on Sentiment Analysis

In recent years, different studies and researchers have proposed several methods for dealing with the new threats of adversarial examples for text applications models. Such defenses aim to deal with modified inputs, seeking to identify and discard them in order to mitigate their negative impact on the model's results. Until now, defensive methods have focused mainly on implementing techniques such as data augmentation and adversarial training or incorporating methods that identify changes in the inputs; these approaches use knowledge of the attack process to intentionally generate adversarial examples that models can learn from to identify and discard possible modifications. Figure 9 illustrates the general operation of actual defenses against adversarial examples.

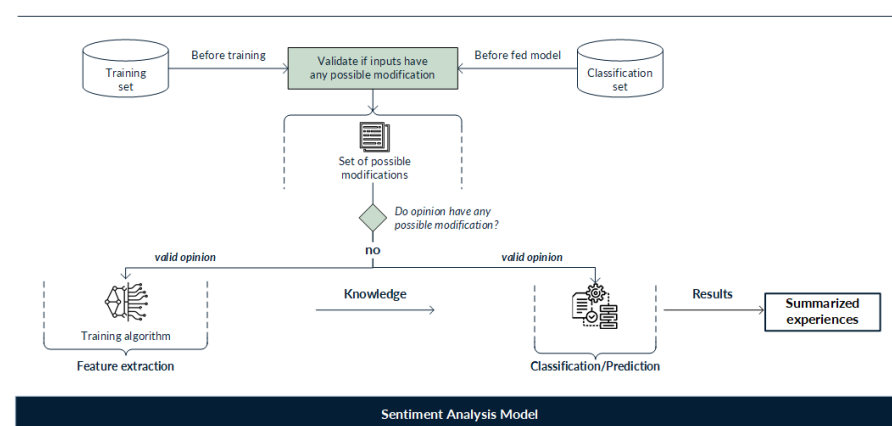


Figure 9. General defense mechanism within a sentiment analysis model. The aim of the defense mechanism is to identify and discard modified inputs in order to mitigate their negative impact on model results.

General functioning of defenses against adversarial examples consists of the following:

1. **Modification control investigates the differences between legitimate and adversarial texts.** Frequently, textual adversarial examples present notable differences compared to the original input texts. For example, adversarial texts may use character-level modifications, in many cases consisting of misspelled words. Consequently, researchers have used spell checkers to identify modified text inputs as a means of detecting this type of adversarial text [40,41]. However, modifications made by substitution with synonyms cannot be detected by these types of defenses.
2. **DNN models can be improved to strengthen them against adversarial attacks.** This defense includes modifying architectures to improve security and working with a training set with known parameters [42].

One fundamental purpose of generating effective adversarial examples is to use them to improve the robustness of existing models [8]. There are two principal strategies in text applications models to achieve this goal, namely, adversarial training and distillation. These strategies are briefly described in the following sections. For more comprehensive reviews of defense strategies for text applications, please refer to [43,44].

4.1. Adversarial Training Strategy

In [7], the authors proposed the adversarial training strategy. This strategy consists of training a neural network to correctly classify both legitimate inputs (inputs without modifications) and adversarial examples. The adversarial strategy includes techniques such as data augmentation, model regularization, and robust optimization, which are explained below.

- **Data augmentation.** This process augments the original training set with the generated adversarial examples and attempts to teach the model to discard them during the training phase.
- **Model regularization.** The model's regularization imposes the adversary examples generated as a form of regularization.

$$\min(J(f(x), y)) + \lambda J(f(x'), y) \quad (7)$$

where λ is a hyperparameter.

- **Robust Optimization.** Madry et al. [45] proposed DNN-model learning as a robustness optimization with a min–max formulation, which is the composition of a non-concave internal maximization problem (attack) and a non-convex external minimization problem (defense).

4.2. Distillation Strategy

Papernot et al. [42] proposed distillation as another defense against adversarial examples. The objective is to use the softmax output of the original neural network (for example, the class probabilities when classification is made) to train a second model with the same structure as the original. The softmax of the original DNN is modified by introducing a temperature parameter T :

$$q_i = \frac{\exp(z_i/T)}{\sum_k \exp(z_k/T)} \quad (8)$$

where z_i is an input of softmax layer and T controls the level of knowledge distillation. When $T = 1$ (refer to Equation (8)) returns to the normal softmax function, if T is large, then q_i is close to a uniform distribution; when it is small, the function will output more extreme values. Grosse et al. [46] adopted distillation defense for discrete data and applied a high temperature T , as a high-temperature softmax has been proven to reduce models' sensitivity to small perturbations.

4.3. Adversarial Defense

In this section, we present the main reviewed defense works proposed against adversarial examples, particularly for sentiment analysis models. These works have been used as a reference for the design of other proposals. Table 2 summarizes the reviewed defenses according to their main characteristics.

Table 2. Reviewed adversarial defenses for sentiment analysis models.

Work	Defense Strategy	Granularity	Defense Evaluation	DNN Model Defended
[47]	Modification control	Character	Accuracy in recognizing misspelled words, Sensitivity to adversarial perturbations on a input	BERT fine-tuned
[40]	Modification control	Term	Model robustnes	CNN, RNN and BERT
[48]	Modification control	Character/Term	Model robustnes	BERT
[34]	Modification control	Term	Model robustnes	CNN, BiLSTM, RNN
[39]	Data augmentation	Term	Model robustnes	CNN, BiLSTM, BERT-Base

- Pruthi et al. [47] proposed designing and implementing a model for word validation and recognition before the classifier. The method is trained to recognize words modified by dropping, adding, and swapping internal characters within words. This defense proposes a two-stage solution to deal with character-level adversarial attacks, placing a word recognition model W before the downstream classifier C . This recognition model is based on a semi-character architecture of RNNs, introducing various feedback strategies for handling uncommon or unseen words. The proposed defense method is inspired by psycholinguistic studies that proposed a semi-character-based RNN (ScRNN) that processes a sentence of words with misspelled characters and predicts the correct words. ScRNN treats the first and the last characters individually and is agnostic about the ordering of the internal characters. At each sequence step, the training target is the correct corresponding word (output dimension equal to vocabulary size), and the model is optimized using the cross-entropy loss.
- Wang et al. [40] proposed a defense mechanism called the Synonym Encoding Method (SEM). This mechanism inserts an encoder before the input layer of the model and then trains the model to remove adversary perturbations. Using an encoder, the defense method substitutes the words in the original sentence with their close synonyms. To define the encoder E that encodes a set of synonyms, the authors clustered the synonyms in the embedding space and allocated a unique token for each cluster, obtaining the same cluster for different words.
- Zhou et al. [48] proposed the DIScriminate Perturbations (DISP) mechanism for identifying and adjusting malicious modifications and blocking attacks. To identify adversarial examples, the discriminator validates the probability of a term in the text being modified, based on which it provides a set of potential modifications. For each potential modification, an insertion estimator learns to restore modified terms by selecting a replacement token based on a search of the k nearest neighbors. The proposed mechanism tries to block adversarial attacks on some models for texts without modifying their structure or the training process.
- Wang et al. [34] proposed a general defense mechanism called TextFirewall for different attacks with different strategies. Given a text input x , TextFirewall analyzes and quantifies each word's impact in the text to distinguish the polarity instead of directly detecting the difference between adversarial text and legitimate input (inputs without any modification). With x feeding the model the $top-k$ items that impact the classification results, if the $loss$ is greater than zero, then y^* is set to 1; otherwise, y^* is 0. The defense mechanism compares y^* and y' to judge the role of x . If y^* is the same as y' , then x is predicted as a legitimate input (input without modification); otherwise, x is an adversarial example.
- Xu et al. [39] presented a gray box adversarial attack and defense framework consisting of a generator ζ (updated) and two copies of a pretrained target classifier: a static classifier C and an updated/augmented classifier C^* . The operation of this method is described above in Section 3.3.

5. Discussion of Adversarial Attacks and Open Challenges for Sentiment Analysis

The design of adversarial example attacks in text applications has become popular in recent years. The volume and depth of contributions regarding defenses for text applications such as sentiment analysis has been less than for other tasks. Therefore, effective attack strategies and defense mechanisms must be explored to ensure the correct functioning of sentiment analysis systems. At present, two critical and significant challenges are present when designing textual adversarial examples: preserving syntax, and validating correct grammar and semantics. Additionally, there are challenges within text applications that inherently need to be addressed within the sentiment analysis task, for example, making modifications that are both imperceptible to humans and effective in confusing models. This might be one of the most challenging problems, as changes within a text are easy to detect even when unintentional, such as orthographic errors. Another remaining challenge

is to ensure the generality of the methods used to create adversarial examples in order to preserve their effectiveness and make them easy to use in other models.

Based on the reviewed literature, this section includes the current challenges in designing adversarial examples:

- **Perceptibility.** While image modifications are often imperceptible to humans, modifications in texts are readily identifiable. Invalid words and syntactical errors can be identified relatively easily using a grammar checking process, allowing them to be discarded. From the point of view of semantic preservation, changing a word in a sentence can drastically change the semantics; thus, without additional processing, the modified inputs can be easily identified and dismissed. To ensure an effective attack, successful approaches must make the modifications imperceptible while preserving correct grammar and semantics [10]. In [49], the authors presented a white box attack method against a word-level CNN text classifier. The proposed approach uses the Euclidean and cosine distances as a combined metric to find the most semantically similar substitution when generating perturbations. In addition, the dispersion of the location of the modified words in the adversarial examples is controlled by introducing a coefficient of variation (CV) factor. Combining these two methods increases the attack success rate and makes the modification positions in the generated examples more dispersed. Recent research seeks to precisely determine the terms that contribute to the correct classification of input and use them to create adversarial examples [32–34]. For sentiment analysis applications, adversarial examples must be carefully designed to consider an effective set of terms that uniquely identify different sentiments (positive, negative, or neutral) that contribute to classifying an opinion.
- **Transferability.** Transferability is a desirable property in adversarial examples, reflecting the generalization of attack methods by ensuring that the adversarial instances created for one model and on one dataset can be used on another model or dataset while remaining effective [10]. In Reference [50], the authors proposed that transferability between seemingly different models is due to a high linear correlation between the feature sets extracted by different networks. In Reference [51], a systematic investigation of factors affecting adversarial examples' transferability for text classification models was explored. The authors contemplated several factors, including network architectures, tokenization schemes, word embedding, and model capacity. On this basis, they proposed a genetic algorithm to find an ensemble of models that can be used to induce adversarial examples to fool different existing models. We remark that to achieve transferability between sentiment analysis models, adversarial examples should be designed under modifications that consider disrupting the characteristics of the task, thereby confounding the model understanding and management of the sentiment analysis task over the understanding of text input.
- **Task-oriented.** Most current works address different tasks by applying global strategies to modify inputs. This does not necessarily provide a correct solution, as specific challenges in each task must be handled to ensure a correct modification process. Although previous adversarial example attacks focusing on sentiment analysis have fooled models and reduced the precision of the results, these works have focused on addressing the sentiment analysis at the document level, and have not modeled the problem to deal with aspect-level natural characteristics. An ideal adversarial example design for sentiment analysis models should combine sentiment analysis and adversarial example characteristics to perform modifications on inputs to achieve task-oriented adversarial examples. In this design, two main issues are encountered. First, to construct task-oriented adversarial examples, it is necessary to correctly determine a set of terms that uniquely identify different sentiments (positive, negative, or neutral) contributing to classifying an opinion. Second, it is necessary to establish the set of possible perturbations N for each term while evaluating and controlling them so that each perturbation can be performed while both preserving the correct semantics and syntax and fooling the model.

- **New architectures.** Several architectures that are widely used in sentiment analysis have not yet been effectively attacked, for example, generative models (Generative Adversarial Networks (GAN)) and Variational Auto-Encoders (VAE). These models require a great deal of experience for model training, which may explain why they have not been effectively attacked. Other architectures that have attracted attention involve attention mechanisms, which are becoming a prevalent component in sequential models. However, as there are few studies examining the functioning of these attention mechanisms, there is a lack of models that can be used to generate modified inputs that are effectively against them. Nevertheless, adversarial examples do not have to identify and exploit DNN vulnerabilities. Instead, the design of adversarial examples should approach the modifications to be performed based on the input elements that support the task. In this way, adversarial attacks could be transferred among different DNN models, at least for the same task.
- **Attack-dependent defenses.** Current defenses against adversarial examples rely on knowledge of the generation process by which the model's inputs were modified, an approach that is not appropriate due to the increasing performance of adversarial examples. More effective defenses must be attack-independent, meaning that they do not require knowledge of the generation process to identify modifications and discard adversarial examples. Notably, such defense mechanisms should be more preventive than reactive.

6. Conclusions

Since their introduction, adversarial examples have pointed out the limitations of deep learning models in correctly classifying intentionally modified inputs. The negative impact of adversarial examples on the deep learning models compels the exploration of new vulnerabilities and defense mechanisms that can effectively cover them to guarantee the model's results.

This work has presented an informative framework in which key concepts concerning the design of adversarial examples are outlined, and has summarized the principal works on attack and defense for deep neural network models performing sentiment analysis tasks. This knowledge is expected to help researchers develop new approaches for designing attacks using adversarial examples and propose new defenses specific to the sentiment analysis task.

At present, designing and implementing adversarial texts remains challenging, as the nature of the data makes it relatively easy to identify modifications in a text. Currently, the main challenge in text applications is to design adversarial examples with imperceptible modifications that preserve the correct semantics and grammar. For effective adversarial examples, design methods must create effectively modified inputs capable of being transferred to other models that can successfully affect the model's results. In addition, it is necessary to explore recent architectures to find vulnerabilities and carry out experiments to find weaknesses in them. On the other hand, due to the effectiveness that adversarial examples have already demonstrated in breaking the safety of deep learning models in the text area, it is necessary to propose effective defense mechanisms to guarantee the reliability of the results. Thus far, the existing defense mechanisms depend primarily on knowledge of the attack process; however, in a real scenario it is not always possible to know this process. Therefore, future defense mechanisms should be independent of the attack mechanism in order to prevent future attacks. We suggest that sentiment analysis models should incorporate defense mechanisms in their design in order to safeguard their data and avoid their results being disturbed.

Author Contributions: Conceptualization, M.V.-H., L.A.M.-R. and I.A.-B.; Data curation, M.-L.C.-T.; Formal analysis, I.A.-B. and H.R.-R.; Investigation, M.V.-H., S.I.F.-G., H.R.-R. and M.-L.C.-T.; Methodology, M.V.-H. and L.A.M.-R.; Software, M.V.-H.; Supervision, L.A.M.-R. and I.A.-B.; Validation, L.A.M.-R., I.A.-B. and H.R.-R.; Visualization, S.I.F.-G.; Writing—original draft, M.V.-H., L.A.M.-R. and I.A.-B.; Writing—review and editing, S.I.F.-G., H.R.-R. and M.-L.C.-T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Mexican National Council of Humanities Science and Technology (CONAHCYT) through scholarships 814461 and 478112 and research projects 882 and 613.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the study's design, in the collection, analysis, and interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AA	After-attack Accuracy
BA	Before-attack Accuracy
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neuronal Network
DL	Deep Learning
DNN	Deep Neuronal Network
GAN	Generative Adversarial Network
IEEE	Institute of Electrical and Electronics Engineers
LSTM	Long Short-Term Memory
BiLSTM	Bidirectional Long Short-Term Memory
NLP	Natural Language Processing
RNN	Recurrent Neuronal Network
SA	Sentiment Analysis
SNLI	Stanford Natural Language Inference
WMD	Word Mover's Distance

References

- Shaba, M. A Real-Time Sentimental Analysis on E-Commerce Sites in Nigeria Using Machine Learning. In Proceedings of the Hybrid Intelligent Systems: 21st International Conference on Hybrid Intelligent Systems (HIS 2021), Online, 14–16 December 2021; p. 452.
- Liu, B.; Zhang, L. A survey of opinion mining and sentiment analysis. In *Mining Text Data*; Springer: Boston, MA, USA, 2012; pp. 415–463.
- Liu, B. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 1.
- Chandrasekaran, G.; Nguyen, T.N.; Hemanth D, J. Multimodal sentimental analysis for social media applications: A comprehensive review. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2021**, *11*, e1415. [\[CrossRef\]](#)
- Zhu, L.; Zhu, Z.; Zhang, C.; Xu, Y.; Kong, X. Multimodal sentiment analysis based on fusion methods: A survey. *Inf. Fusion* **2023**, *95*, 306–325. [\[CrossRef\]](#)
- Sarker, I.H. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *Comput. Sci.* **2021**, *2*, 420. [\[CrossRef\]](#) [\[PubMed\]](#)
- Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
- Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2014**, arXiv:1412.6572.
- Jia, R.; Liang, P. Adversarial Examples for Evaluating Reading Comprehension Systems. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 9–11 September 2017; pp. 2021–2031. [\[CrossRef\]](#)
- Zhang, W.E.; Sheng, Q.Z.; Alhazmi, A.; Li, C. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol.* **2020**, *11*, 1–41. [\[CrossRef\]](#)
- Meng, D.; Chen, H. Magnet: A two-pronged defense against adversarial examples. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 135–147.
- Alsmadi, I.; Ahmad, K.; Nazzal, M.; Alam, F.; Al-Fuqaha, A.; Khreishah, A.; Algosaiibi, A. Adversarial attacks and defenses for social network text processing applications: Techniques, challenges and future research directions. *arXiv* **2021**, arXiv:2110.13980.
- Xu, H.; Ma, Y.; Liu, H.C.; Deb, D.; Liu, H.; Tang, J.L.; Jain, A.K. Adversarial attacks and defenses in images, graphs and text: A review. *Int. J. Autom. Comput.* **2020**, *17*, 151–178. [\[CrossRef\]](#)

14. Han, X.; Zhang, Y.; Wang, W.; Wang, B. Text Adversarial Attacks and Defenses: Issues, Taxonomy, and Perspectives. *Secur. Commun. Netw.* **2022**, 2022, 6458488. [\[CrossRef\]](#)
15. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, 35, 1798–1828. [\[CrossRef\]](#)
16. Gong, Z.; Wang, W.; Li, B.; Song, D.; Ku, W.S. Adversarial Texts with Gradient Methods. *arXiv* **2018**, arXiv:1801.07175.
17. Tsai, Y.T.; Yang, M.C.; Chen, H.Y. Adversarial Attack on Sentiment Classification. In Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Florence, Italy, 1 August 2019; pp. 233–240. [\[CrossRef\]](#)
18. Li, J.; Ji, S.; Du, T.; Li, B.; Wang, T. TextBugger: Generating Adversarial Text Against Real-world Applications. In Proceedings of the 2019 Network and Distributed System Security Symposium, San Diego, CA, USA, 24–27 February 2019. [\[CrossRef\]](#)
19. Alzantot, M.; Sharma, Y.; Elgohary, A.; Ho, B.J.; Srivastava, M.; Chang, K.W. Generating Natural Language Adversarial Examples. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2890–2896. [\[CrossRef\]](#)
20. Yuan, X.; He, P.; Zhu, Q.; Li, X. Adversarial examples: Attacks and defenses for deep learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, 30, 2805–2824. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Wang, T.; Zhu, L.; Zhang, Z.; Zhang, H.; Han, J. Targeted Adversarial Attack Against Deep Cross-Modal Hashing Retrieval. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, 33, 6159–6172. [\[CrossRef\]](#)
22. Eger, S.; Şahin, G.G.; Rücklé, A.; Lee, J.U.; Schulz, C.; Mesgar, M.; Swarnkar, K.; Simpson, E.; Gurevych, I. Text Processing Like Humans Do: Visually Attacking and Shielding NLP Systems. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Long and Short Papers, Volume 1, pp. 1634–1647. [\[CrossRef\]](#)
23. Zang, Y.; Qi, F.; Yang, C.; Liu, Z.; Zhang, M.; Liu, Q.; Sun, M. Word-level Textual Adversarial Attacking as Combinatorial Optimization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6066–6080. [\[CrossRef\]](#)
24. Gan, W.C.; Ng, H.T. Improving the Robustness of Question Answering Systems to Question Paraphrasing. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 6065–6075. [\[CrossRef\]](#)
25. Liang, B.; Li, H.; Su, M.; Bian, P.; Li, X.; Shi, W. Deep Text Classification Can be Fooled. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18. International Joint Conferences on Artificial Intelligence Organization, Stockholm, Sweden, 13–19 July 2018; Volume 7, pp. 4208–4215. [\[CrossRef\]](#)
26. Vijayaraghavan, P.; Roy, D. Generating Black-Box Adversarial Examples for Text Classifiers Using a Deep Reinforced Model. In *Machine Learning and Knowledge Discovery in Databases*; Springer International Publishing: Cham, Switzerland, 2020; pp. 711–726. [\[CrossRef\]](#)
27. Zhang, J.; Li, C. Adversarial examples: Opportunities and challenges. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, 31, 2578–2593. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Glasgow, UK, 2014; pp. 3104–3112.
29. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473.
30. Quintero, Y.C.; Garcia, L.A. Estudio del análisis de sentimientos basado en espectos. In Proceedings of the “IV Conferencia Internacional en Ciencias Computacionales e Informáticas”, Havana, Cuba, 26–30 November 2018.
31. Graves, A. Long short-term memory. In *Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 37–45.
32. Ma, Y.; Peng, H.; Cambria, E. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
33. Xiao, Y.; Zhou, G. Syntactic edge-enhanced graph convolutional networks for aspect-level sentiment classification with interactive attention. *IEEE Access* **2020**, 8, 157068–157080. [\[CrossRef\]](#)
34. Wang, W.; Wang, R.; Ke, J.; Wang, L. TextFirewall: Omni-Defending Against Adversarial Texts in Sentiment Classification. *IEEE Access* **2021**, 9, 27467–27475. [\[CrossRef\]](#)
35. Ribeiro, M.T.; Singh, S.; Guestrin, C. Semantically Equivalent Adversarial Rules for Debugging NLP models. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Long Papers; Volume 1, pp. 856–865. [\[CrossRef\]](#)
36. Gao, J.; Lanchantin, J.; Soffa, M.L.; Qi, Y. Black-box generation of adversarial text sequences to evade deep learning classifiers. In Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 24 May 2018; pp. 50–56.
37. Jin, D.; Jin, Z.; Zhou, J.T.; Szolovits, P. 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, New York, NY, USA, 7–12 February 2020; Volume 34; pp. 8018–8025.
38. Ekbal, A. Adversarial Sample Generation for Aspect based Sentiment Classification. In Proceedings of the Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022, Online, 20–23 November 2022; pp. 478–492.

39. Xu, Y.; Zhong, X.; Jimeno Yepes, A.; Lau, J.H. Grey-box Adversarial Attack and Defence For Sentiment Classification. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; pp. 4078–4087. [\[CrossRef\]](#)
40. Wang, X.; Jin, H.; Yang, Y.; He, K. Natural Language Adversarial Defense through Synonym Encoding. In Proceedings of the Conference on Uncertainty in Artificial Intelligence, Online, 27–30 July 2021.
41. Wang, Z.; Wang, H. Defense of word-level adversarial attacks via random substitution encoding. In Proceedings of the International Conference on Knowledge Science, Engineering and Management, Hangzhou, China, 28–30 August 2020; pp. 312–324.
42. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 582–597.
43. Goyal, S.; Doddapaneni, S.; Khapra, M.M.; Ravindran, B. A survey of adversarial defenses and robustness in nlp. *ACM Comput. Surv.* **2023**, *55*, 1–39. [\[CrossRef\]](#)
44. Zhou, Y.; Zheng, X.; Hsieh, C.J.; Chang, K.W.; Huang, X. Defense against adversarial attacks in nlp via dirichlet neighborhood ensemble. *arXiv* **2020**, arXiv:2006.11627.
45. Mađry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *Stat* **2017**, *1050*, 9.
46. Grosse, K.; Papernot, N.; Manoharan, P.; Backes, M.; McDaniel, P. Adversarial examples for malware detection. In Proceedings of the Computer Security—ESORICS 2017: 22nd European Symposium on Research in Computer Security, Oslo, Norway, 11–15 September 2017; Part II 22; Springer: Cham, Switzerland, 2017; pp. 62–79.
47. Pruthi, D.; Dhingra, B.; Lipton, Z.C. Combating Adversarial Misspellings with Robust Word Recognition. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019. [\[CrossRef\]](#)
48. Zhou, Y.; Jiang, J.Y.; Chang, K.W.; Wang, W. Learning to Discriminate Perturbations for Blocking Adversarial Attacks in Text Classification. *arXiv* **2019**, arXiv:1909.03084.
49. Du, X.; Yi, Z.; Li, S.; Ma, J.; Yu, J.; Tan, Y.; Wu, Q. Generating More Effective and Imperceptible Adversarial Text Examples for Sentiment Classification. In Proceedings of the International Conference on Artificial Intelligence and Security, Dublin, Ireland, 19–23 July 2020; pp. 422–433.
50. Wiedeman, C.; Wang, G. Disrupting adversarial transferability in deep neural networks. *Patterns* **2022**, *3*, 100472. [\[CrossRef\]](#) [\[PubMed\]](#)
51. Yuan, L.; Zheng, X.; Zhou, Y.; Hsieh, C.J.; Chang, K.W. On the Transferability of Adversarial Attacks against Neural Text Classifier. In Proceedings of the Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.