Adversarial Examples and Defense Mechanisms for Aspect-Based Sentiment Analysis Models

Technical Report: CCC-23-003.

by

Monserrat Vázquez Hernández

Doctoral Advisors:

Dr. Ignacio Algredo Badillo, INAOE

Dr. Luis Villaseñor Pineda, INAOE

Instituto Nacional de Astrofísica, Óptica y Electrónica

May, 2023

Santa María de Tonantzintla, Puebla, CP 72840

Contents

Abstract

In recent years, the use of Deep Learning models for deploying Sentiment Analysis systems at aspect-level has become a widely-used topic due to their processing capacity and superior results achieved on large volumes of information. However, after several research years, previous works have demonstrated that Deep Learning models are vulnerable to strategically modified inputs named adversarial examples. Adversarial examples are modified inputs generated by performing imperceptible perturbations to humans in data, which are able to fool Deep Learning models and generate incorrect results. Previous adversarial works focusing on sentiment analysis have shown to effectively fool models and cause incorrect results. Nevertheless, these works lack of proper modeling the aspect-level analysis which has led to perform irrelevant perturbations impacting on modifications imperceptibility and message's readability. Besides, the proposed defenses against adversarial examples have been dependent on process knowledge by which the inputs were modified to try to identify and discard them. In this work, we propose to define a model for adversarial-examples generation, especially suited to sentiment analysis at aspect-level in which level's characteristics are considered to drive the course of modifications. Additionally, this work aims to propose attack-independent defenses whose operation does not rely on the input-modification process trying to guarantee the authenticity and integrity of inputs. According to the objectives and the established methodology, we have been experimenting on define a new model for aspect-level adversarial-example generation considering task's characteristics on a white-box scenario. We evaluate our proposal against baseline adversarial examples generated via document-level strategies, the obtained results show the effectiveness of our proposal overcoming document-level strategies in a 12.8% making target model's accuracy drops 20.9% and maintaining the modification' imperceptibility according to semantic similarities of adversarial examples in a 99.0% concerning to original inputs. This preliminary work shows promising results for effectively modeling aspect-level adversarial examples maintaining imperceptible perturbations and message's readability. Hence, we will characterize input modifications and evaluate the proposed model on different scenarios and architectures.

1 Introduction

Digital opinions allow to users and organizations to identify the experience, positive or negative, that actual users have had about a product, service or topic of interest. Thanks to user's opinion, new potential consumers can look at the quality of a product or service and thus, decide whether or not to select it. Besides, knowing users opinions allow organizations to determine the necessary improvements to implement in their products or services to enhance their actual users' experience. In this context, sentiment-analysis systems are an important tool that provides summarized information of users opinions to assist potential users and organizations to evaluate their selections. sentiment analysis (SA) concerns to the use of text analysis and machine-learning techniques to the automatic extraction and processing of users opinions (Liu, 2011). Through SA systems, it is attempted to determine a user experience of a product or service based on the positive or negative connotation of the language used to express its opinions. Figure 1 illustrates the general scheme operation of a sentiment analysis system.

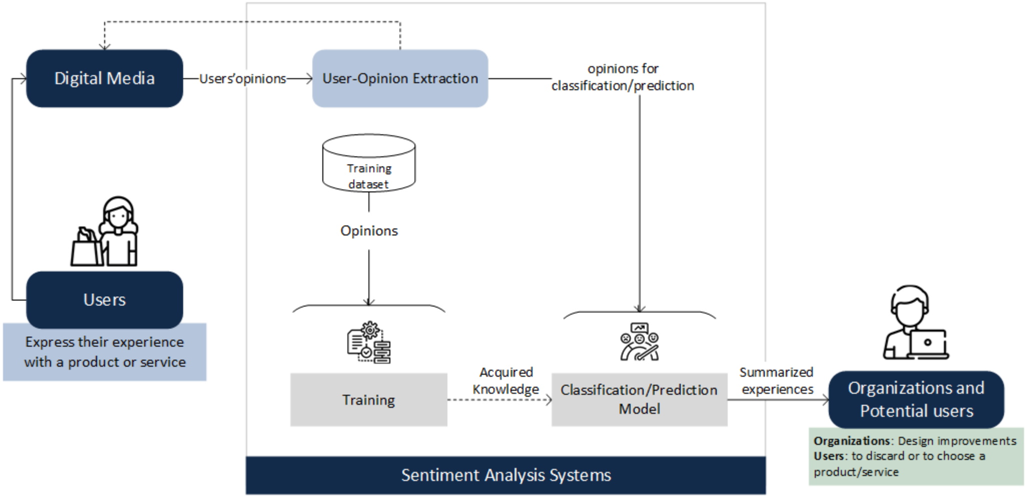
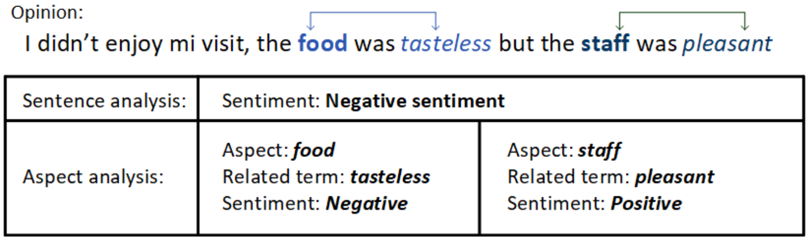

Digital Media
Users'opinions
User-Opinion Extraction
opinions for classification/prediction
Training dataset
Opinions
Users
Express their experience with a product or service
Training
Acquired Knowledge
Classification/Prediction Model
Summarized experiences
Organizations and Potential users
Organizations: Design improvements
Users: to discard or to choose a product/service
Sentiment Analysis Systems

Figure 1: Overview of a general scheme operation of a sentiment analysis system.

According to information needs, the sentiment analysis can be performed at different granularity levels: i) document, ii) sentence or iii) aspect[1]. The analysis at document-level refers to the positive or negative classification of a full text (Pang et al., 2002) while in analysis at sentence-

[1] The term "aspect" is used to name components, characteristics or attributes of a product, service or entity.

level, the objective is to analyze each sentence in a text in order to independently classify them (Riloff and Wiebe, 2003). Finally, the analysis at aspect-level (or aspect-based sentiment analysis ABSA) seeks to independently determine the opinion expressed for each mentioned aspect within an opinion (Poria et al., 2020).



Opinion:

I didn't enjoy mi visit, the **food** was *tasteless* but the **staff** was *pleasant*

Sentence analysis:	Sentiment: **Negative sentiment**	
Aspect analysis:	Aspect: ***food*** Related term: ***tasteless*** Sentiment: ***Negative***	Aspect: ***staff*** Related term: ***pleasant*** Sentiment: ***Positive***

Table 1: Sentiment analysis at aspect-level. In this level, the user attitude is determine for each mentioned aspect within opinion.

In table 1, it is illustrated the differences when it is performing the analysis at sentence and at aspect-level. By nature, different aspects can be included within an opinion and, for each of them, different attitudes can be expressed. So, to determine a user's opinion towards aspects it is necessary to identify the aspect-terms relation; that is to say, to identify the correspondent opinion terms related to each aspect and thus determine the positive o negative user opinion by each one. In many cases, the sentiment analysis at document or sentence level does not provide specific details about particular aspects. For example, in a document with positive opinions about an entity does not mean that the user has positive opinions about all aspects and, similarly, a negative document about an entity does not mean that the user has negative opinions about all aspects of this entity (Liu and Zhang, 2012), given this situation, it is necessary to work at a lower granularity, hence the interest and importance of aspect-level analysis.

Aspect-level analysis, being a more detailed task, requires methods that accurately identify the opinion-terms related to each evaluated aspect to provide accurate information about current users attitudes. In last years, the use of Deep Learning (DL) models to address the aspect-level analysis has gained great popularity; through DL models it is pursued to improve the precision of results and increase the confidence of its users, although this does not always turn out to be true. After several years, different research works have demonstrated that DL models can be fooled with high probability by strategically-modified inputs denominated as adversarial examples.

Adversarial examples are modified inputs generated to cause a negative impact on models' results. The adversarial examples are generated by adding some small and subtle modifications to original inputs to confuse the models on inputs' understanding and thus cause their incorrect classification (according to the classification task). Formally, an adversarial example x' is a modified input created via a perturbation n of the input x to a DL model. The perturbation n is the minimal worst-case modification to input data which succeeds in confusing the model in its classification. A robust DL model should continue to classify the correct class y to x', while a victim model would have a high probability of the wrong classification of x' (Zhang et al., 2020). The aim of adversarial examples x' is deviating the correct label to incorrect one $f(x') \neq y$ or to an specific one $f(x') = y'$. In equation 1 it is expressed the global adversarial examples formalization:

$$\begin{gathered} f(x) = y,\ x \in X \qquad (1) \\ x' = x + n,\ f(x') \neq y \\ sor f(x') = y',\ y' \neq y \end{gathered}$$

Szegedy et al. (2013) introduced adversarial examples when they studied the stability of state-of-the-art Deep Neural Networks (DNNs) for image classification in face of modified inputs. In their work, they performed small pixel-level modifications to input data and observed that DNNs could be fooled by these modified inputs even human perception of data is not affected (Zhang et al., 2020). Based on adversarial example idea, Jia and Liang (2017) are the first to consider the adversarial-example design to evaluate DNNs models for a text-based task. In their work, they experimented by inserting text fragments at the end of inputs, without change the original text and they observed that DNNs text-models could also be fooled by adversarial examples.

Generating adversarial examples is motivated by one of two objectives: attack or defense. Adversarial attacks aim to examine the robustness of target model, while defenses use the adversarial-example knowledge to strengthen models (Zhang et al., 2020). An adversarial attack consists of generating and inserting adversarial examples into input data model to compromise its results. According to the knowledge about the model to be fooled (or victim model), three types of attacks can be carried out: *White box*, *Black box* and *Grey box*. In figure 2 it is illustrated the general methodology of adversarial attacks to a sentiment analysis system according to model's knowledge (a victim model can suffer attacks under different levels of knowledge at same time, however, in this work we will focus on studying the attacks independently).

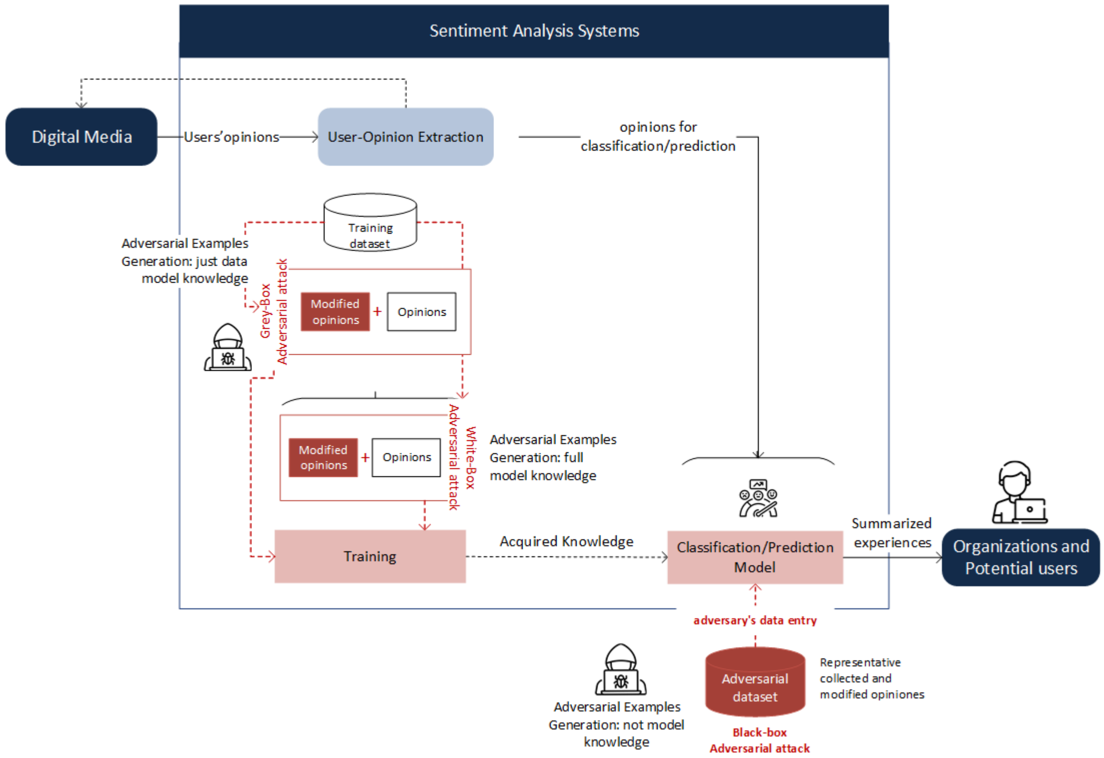
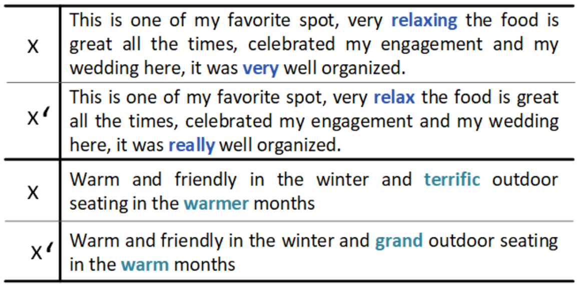

Sentiment Analysis Systems
Digital Media
Users'opinions
User-Opinion Extraction
opinions for classification/prediction
Training dataset
Adversarial Examples Generation: just data model knowledge
Grey-Box Adversarial attack
Modified opinions
+
Opinions
Modified opinions
+
Opinions
White-Box Adversarial attack
Adversarial Examples Generation: full model knowledge
Training
Acquired Knowledge
Classification/Prediction Model
Summarized experiences
Organizations and Potential users
adversary's data entry
Adversarial dataset
Representative collected and modified opiniones
Adversarial Examples Generation: not model knowledge
Black-box Adversarial attack

Figure 2: General methodology of an adversarial attack for a sentiment analysis model.

On the one hand, the white-box attacks rely on knowledge of the complete details of the target model to be fooled including architecture, parameters, activation and loss functions, and input and output data. White-box attacks approximate a worst-case attack on a particular model and inputs incorporating a set of perturbations (Zhang et al., 2020). Typically, white-box attacks make modifications to training data in order to make the model incorrectly learns features about users opinions and thus produce incorrect results. On the other hand, the 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 applying heuristics in a local model (to represent the original victim model) which is trained until modifications that allow changing results are founded. Usually, black-box collects representative data to generate adversarial examples and later, the adversarial examples generated via substitute model are introduced to the victim model to cause incorrect results. Finally, the gray-box attacks are at a middle ground between black-box and white-box attacks. Generally, gray-box attacks have knowledge of the input data of the victim model but not of its technical details.

Different works oriented to text-based tasks have shown that performing modifications at character, term or sentence level by inserting, deleting, substituting or exchanging characters or terms, it is possible to cause incorrect models' results (Gong et al., 2018; Tsai et al., 2019; Li et al., 2018; Alzantot et al., 2018). In table 2, modified text-inputs by substituting a term by its synonym are presented. To generated adversarial examples, the modifications must be as small as possible but capable of fooling models, furthermore, for text-based tasks the modifications should not make drastic changes in the text' semantics and syntax as well as maintain the readability of the input message.



x	This is one of my favorite spot, very **relaxing** the food is great all the times, celebrated my engagement and my wedding here, it was **very** well organized.
x'	This is one of my favorite spot, very **relax** the food is great all the times, celebrated my engagement and my wedding here, it was **really** well organized.
x	Warm and friendly in the winter and **terrific** outdoor seating in the **warmer** months
x'	Warm and friendly in the winter and **grand** outdoor seating in the **warm** months

Table 2: Adversarial examples with a synonym term change. x represent the original input and x' its adversarial example generated. In bold are indicated the modified terms.

A key purpose of generating adversarial examples is to be able to use them to improve the robustness of models (Goodfellow et al., 2014). In recent years, several studies and researchers are proposing different methods to deal with the new threats of adversarial examples for text-applications models. The aim of defenses is to deal with modified inputs to identify and discard them to mitigate their negative impact on model's results. Until now, the defense methods have focused mainly on implementing techniques such as data augmentation, adversarial training or incorporating methods that identify changes in the inputs; these approaches use knowledge of the attack process to intentionally generate adversarial examples to models learn from them to identify possible modifications and discard them. Figure 3 illustrates the general operation of actual defenses by means of adversarial examples.

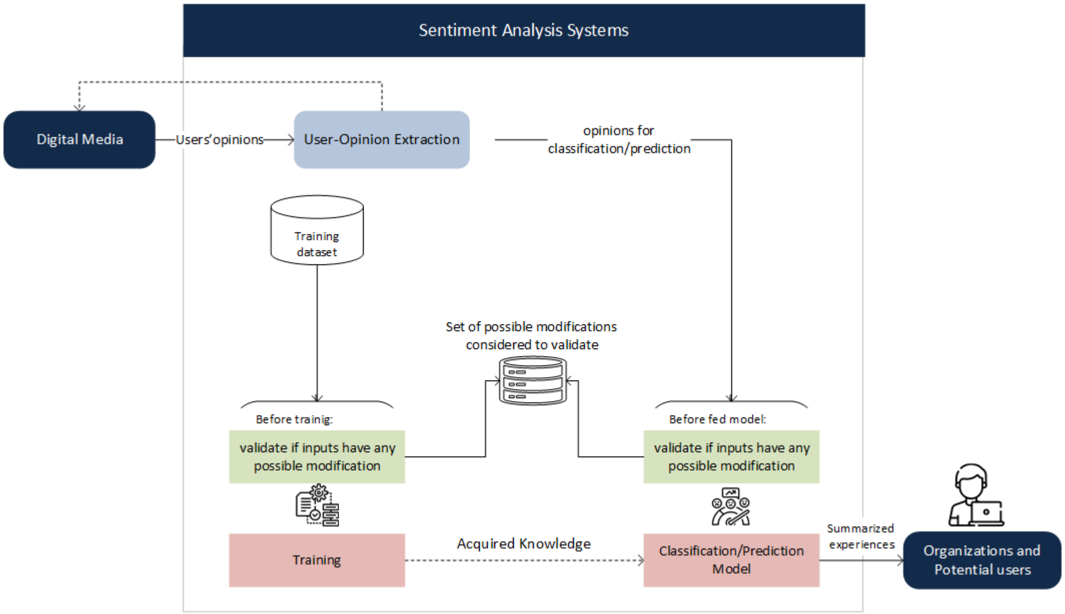

Sentiment Analysis Systems
Digital Media
Users' opinions
User-Opinion Extraction
opinions for classification/prediction
Training dataset
Set of possible modifications considered to validate
Before trainig:
validate if inputs have any possible modification
Before fed model:
validate if inputs have any possible modification
Training
Acquired Knowledge
Classification/Prediction Model
Summarized experiences
Organizations and Potential users

Figure 3: General operation of defenses against adversarial examples.

Since their introduction, adversarial examples have pointed out the limitations of deep learning models to correctly classify modified inputs (Meng and Chen, 2017), these limitations have attracted the interest for identifying models' vulnerabilities in face of modified inputs and determine defenses to guarantee correct model's results. At present, aspect-level systems are an important tool for users and organizations to evaluate their decisions; this led us to increase the research efforts to improve the safety of models' results by means of adversarial examples to avoid potentially negative consequences in scenarios in which sentiment analysis systems are implemented.

Considering the potential of adversarial examples to have a negative impact on the models, it is necessary to first understand how they work and identify vulnerabilities not yet exploited in order to propose defenses to cover them to avoid negative impact on results of deep learning models. In this context, in this project we focus on define a model to generate adversarial examples particularly oriented to aspect-based sentiment analysis to identify and exploit their vulnerabilities and later to propose adversarial defenses to cover them to guarantee correct results. The remainder of this thesis proposal document is organized as follows: in following sections the identified problem is posed. In Section 2, our research proposal is presented, at same time, the hypothesis, objectives, expected contribution, justification, methodology and the proposed activities program schedule are presented. Section 3 includes the background knowledge of this project. In Section 4 the related work and state-of-the-art is described. Finally, in Section 5 the preliminary experiments and results are discussed and in Section 6 the final remarks and work directions are presented.

1.1 Problem Statement

Until now, adversarial-example design for text-applications has problems concerning to: i) on attacks side, to consider task's characteristics to model the adversarial design and, ii) on defense side, adversarial defenses have a clear dependence on attack process knowledge to identify modified inputs and discard them.

1.1.1 Adversarial attacks

Up to now, most current works address different tasks applying global strategies to modify inputs, which does not necessarily provide a correct solution since there are specific challenges in each task that must be handled for a correct modification process. Although previous adversarial-examples attacks focusing on sentiment analysis have shown to fool models and reducing the precision of the results, these works have just focused on addressing the sentiment analysis at document-level and, to the best of our knowledge, they have not modeled the problem to dealt with aspect-level characteristics. At aspect-level analysis, it is crucial to correctly identify the opinion terms which define the user's attitude concerning to an specific aspect.

The lacking of particularized methods to aspect-level has led to modify irrelevant opinion terms that are not related to evaluated aspects making that current methods validate an exponential number of possible modifications as well as increase in the number of modifications which could generate syntactically-incorrect opinion without accomplish the desired effect of adversarial examples, impacting on the imperceptibility of modifications and the input's readability. By other hand, the key factor for a practical adversarial-attack is its transferability from one model to other and still be as effective in fooling them but, due to the deficiency of particularized problem modeling, it has prevented the effective transfer of adversarial attacks among models even when they are for the same task.

To sum it all up, we conclude that an ideal adversarial-example design for analysis at aspect-level has to combine aspect-terms relation and adversarial examples characteristics to perform modifications on inputs achieving task-oriented adversarial examples. In this design, there are two main issues that will be facing. First, to accomplish aspect-level adversarial examples it will be necessary to correctly determine aspect-terms relation a challenging problem which will include complexity to attack, moreover, to set aspect-terms will be determined the selection process to filter opinion terms. Second, for each term in aspect-terms relation it will be necessary to establishing the set of possible perturbations N evaluating and controlling that each one could be performed to the aspect-term relation be infringed but taking care of preserving the correct semantics and syntax and successfully fool models. To address the issues, an algorithm attack must be designed to maintain at minimum the computational complexity to ensure their viable implementation.

1.1.2 Adversarial defenses

The use and availability of successful adversarial attacks revel the need for effective defense methods. Until now, the existing defense have mainly focused on applying strategies as data augmentation, input preprocessing and iterative methods (Zhou et al., 2019; Wang and Wang, 2020; Wang et al., 2021a). Although these defense have helped to care model's results, they have a dependence on attack process knowledge.

Started with data augmentation, defense methods intentionally create and incorporate adversarial examples into dataset to make model learn to identify and discard them. Unfortunately, these methods rely on the idea that models could be attacked under specific modifications and usually do not resist when different ones are performed. For example, defenses against term-level attacks, in which only a list of "most important words" that could be modified (e.g. by its synonym) are safeguarded without considering that there could be an attack for other terms with a different technique. In a real scenario, these approaches are not feasible, because it is not possible to know the process by which the inputs have been modified. For its part, input preprocessing defenses methods require inserting an step between the input data and the given model to identify any possible modification and, in the same way, to discard those inputs that are considered as modified, similar to data augmentation, preprocessing methods rely on the knowledge of the attack process and do not resist in the face of different modifications. Although the search of modifications in text-inputs could be increased to cover more possibilities, this possibility represents a problem in the resources needed to process the information as well as the possible over-fitting of the model when intentionally adversarial examples well know are incorporated. Finally, iterative methods validate if certain modifications negatively impact on model's results based on the output gradient. Conversely, if the modifications are ineffective, the strategy for making modifications in the original data must be updated. The adversarial examples created under these methods show high quality and effectiveness, making disturbances small enough and difficult to defend. However, these methods often take a long time to find the right modifications to make, which is a problem for real-time attacks. Although this problem has been explored in other fields such as computer vision (Liu et al., 2022), exploring potential solutions in the text application area is necessary.

2 Research Proposal

To propose effective defenses for aspect-level models, firstly, it is necessary to examine and identify vulnerabilities on models by means of adversarial examples. So that, it is necessary to define strategies to perform modifications which accomplish the desired effect of adversarial examples maintaining the imperceptibility of modifications, the legibility of inputs and achieve their transferability. Once vulnerabilities have been identified, it is necessary to propose effective defenses to guarante the correct accuracy of models.

2.1 Adversarial attacks

An ideal adversarial-example design for aspect-level model, has to combine aspect-level and adversarial examples characteristics. Based on this, we have modeled in a particularized way the generation of aspect-level adversarial examples as follow: Given an opinion x consisting of n terms $x = \{t_1, t_2, ..., t_n\}$ with m different aspects mentioned $asp = \{asp_1, asp_2, ..., asp_m\}$. For each aspect asp_i there area different terms $t \in x$ particularly related to them $t_{asp_i} = \{t_{si}, t_{si} + l_i\}$ (l_i is the number or words in t_{asp_i} which express its ground truth sentiment y_{asp_i} which should be understood and classified by the model M:

$$M(x, asp_i, t_{asp_i}) = y_{asp_i} \tag{2}$$

The goal of aspect-level adversarial examples, is to generate an adversarial example x' via the modification of t_{asp_i} generating t'_{asp_i} and causing that M performs asp_i misclassification:

$$M(x', asp_i, t'_{asp_i})! = y_{asp_i} \tag{3}$$

At same time, x' should satisfy the following properties:

- Terms in t_{asp_i} can be uni-gram or n-gram words. To set t_{asp_i}, the proximity between aspect asp_i and terms within x have to be computed; this proximity can be expressed as:

$$prox(a_i, t_i) = [0, 1] \tag{4}$$

A $prox(asp_i, t_i) \approx 0$ will be mean that t_i is not related to asp_i while $prox(a_i, t_i) \approx 1$ indicates a relation between t_i and a_i. So far, the $prox$ is calculated via cosine distance.

- To generate t'_{asp_i} each possible modification to $tint_{asp_i} = \{t_{si}, t_{si} + l_i\}$ should maintain the proximity to the original term, i.e $prox(t_i, t'_i) \approx 1$

- The modified input x' should be semantically similar to x. For this, $prox(t, t')$ is calculated via cosine distance between x and x'.

The main challenge to accomplish aspect-level adversarial examples, it is to design an algorithm to effectively combine aspect-terms property and adversarial examples characteristics seeking to maintain at minimum the complexity and achieve a change in input classification. Additionally, it will require an extensive evaluation of the designed algorithm attack to evaluate its effectiveness when it is transferred among different target models considering different knowledge's levels of technical details of them. What is more, this evaluation has to be carried out on different architectures, such as transformers as BERT, to observe its impact on new architectures. In figure 4, the methodology of our aspect-oriented attack is shown in a generalized way, according to the level of knowledge of target model, the modified data can be inserted in the training phase or in the classification/prediction.

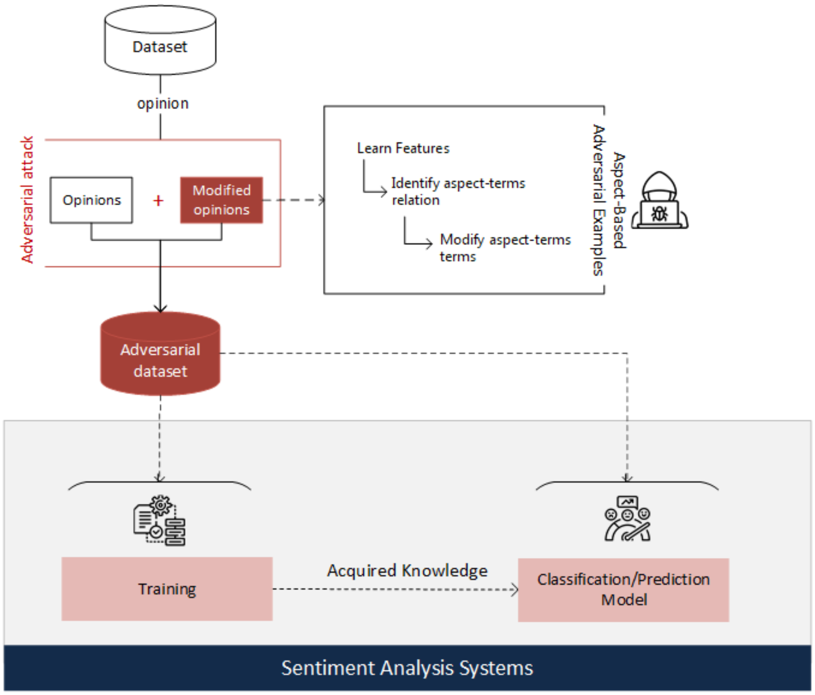


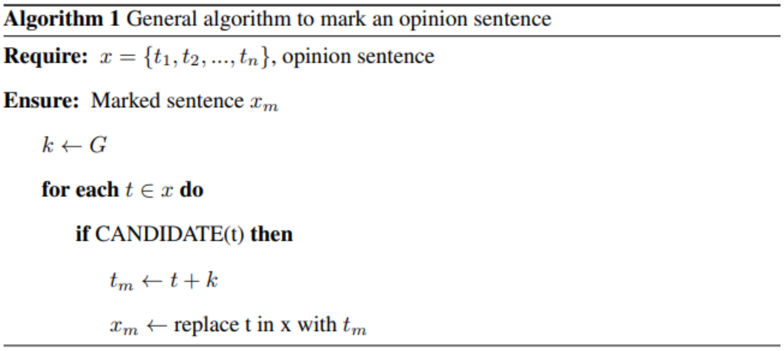
Figure 4: General methodology of defenses aspect-based adversarial attack.

2.2 Adversarial defenses

The key factor for a practical defense against adversarial examples is the generalization. In contrast to actual proposed methods, defense against adversarial examples must be attack-independent, and, rather than attempting to identify adversarial-examples from an specific generation process, defenses would be more anticipatory. The reliance on attack process knowledge has as a consequences the reduction of the generality and, a decrease in their effectiveness in guaranteeing correct model's results. Instead of finding adversarial modifications from an specific generation process, a defense would be more generally by setting a validation process to guaranteeing that model's input have not been modified by a third party.

We consider that effective defenses should not necessarily require the knowledge of modification process but should focus on validating the authenticity of inputs before they are introduced to model and, in this way, guaranteeing they have not been modified, an approach that has not been proposed to date; those inputs which have not been identified as authentic or their integrity is questionable can be discarded. To validate that inputs come from a known source and have not been modified; the message authentication allow to protect the integrity of input as well as its integrity. Message authentication can be seen as a composition of three efficient algorithms (G, S, V) satisfying:

- G (key-generator). A key generation algorithm generates a key (based on key space) uniformly at random: $k \leftarrow G(1^n)$ for each opinion x.
- S (signing). A signing algorithm returns x marked with its previously generated key. Algorithm 1 presents, in a general way, our proposal to mark an opinion:



Algorithm 1 General algorithm to mark an opinion sentence

```
Require: $x = \{t_1, t_2, ..., t_n\}$, opinion sentence
Ensure: Marked sentence $x_m$
  $k \leftarrow G$
  for each $t \in x$ do
    if CANDIDATE(t) then
      $t_m \leftarrow t + k$
      $x_m \leftarrow$ replace t in x with $t_m$
```

CANDIDATE function have to be properly defined, we assumed that to give it robustness this function will validate if t it is related to the evaluated aspect and to add dynamism it will be decided whether or not the term will be marked.

- V (verifying). A verifying algorithm verifies the authenticity of the message given the key and the input x. That is, return accepted when the input has the key and otherwise return rejected.

$$Pr[\, k \leftarrow G(1^n), V(k, x, S(k, x)) = accepted \,] = 1. \tag{5}$$

To protect models against adversarial examples, message authentication algorithm have to be designed to, first, effectively add the given key to each input and then efficiently validate that given the key-space and the text-input, it has not been modified. In figure 5, it is illustrated the general operation of the proposed preventive defense. This algorithm have to be evaluated as a defense against adversarial examples to maintain the accuracy of model. Since the defense has to present generality, it is necessary to evaluate it against different adversarial attacks for distinct aspect-level models.

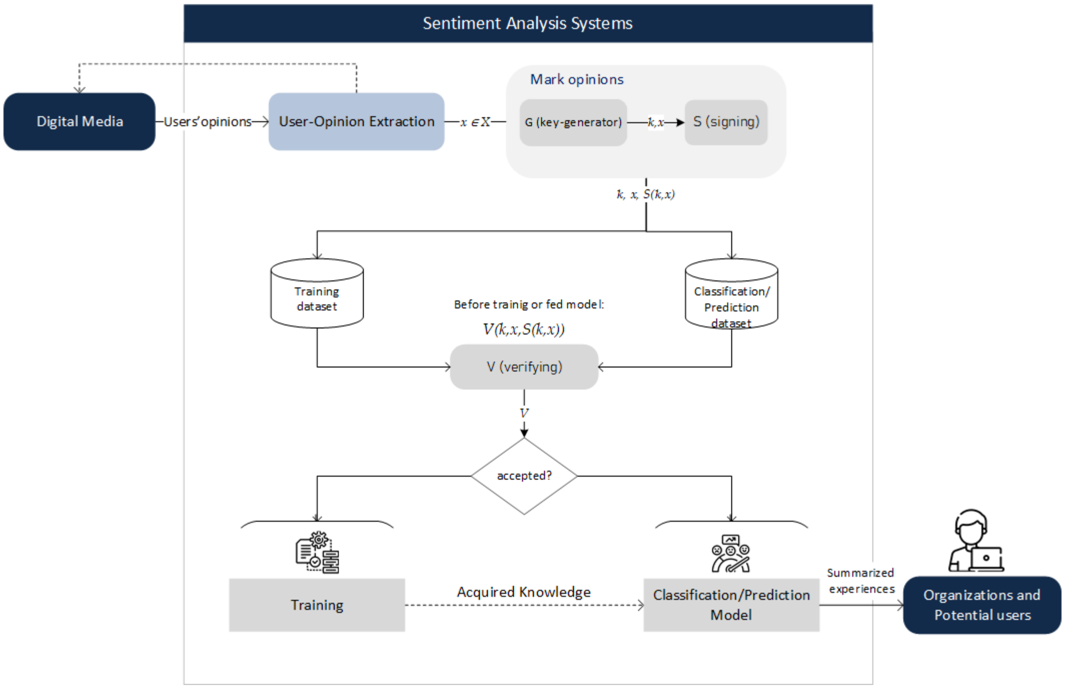


Figure 5: General methodology of proposed preventive defense.

2.3 Research Questions

The questions driving this research work, concerning to attacks and defenses, are as follow:

Q1: Do aspect-level adversarial examples allow to design adversarial attacks to negative impact on correct aspect-level models' results?

Q2: Do aspect-level adversarial examples permit to create modified inputs in which is improved the imperceptibility of modifications, the readability of inputs and to allow the effective transfer of the attack?

Q3: Determining the authenticity and integrity of text-inputs models allow to generate a preventive defense against adversarial examples (with out rely on the knowledge of modification process) and maintain correct models' results?

Q4: A preventive defense based on authenticity and integrity could maintain the correct models' results against aspect-level adversarial examples?

2.4 Hypothesis

If it is considered that adversarial attacks can be used to identify vulnerabilities and thus propose defenses, or alternatively, that defenses use the knowledge of vulnerabilities identified to design adversarial attacks. Our hypothesis under an integral solution indicates:

H: By determining the authenticity and integrity of text-input data allow to define attack-independent defenses against aspect-level adversarial examples allowing to preserve correct model's results for sentiment analysis at aspect-level.

Our supporting hypotheses for attacks and defenses indicate:

$\mathbf{H}_{attacks}$: By focusing on aspect-term relation, modifications to generate adversarial examples will be performed on the minimum necessary terms that effectively support aspect-sentiment which will contribute to perform the fewer modifications, maintaining the imperceptibility of modifications, the readability of inputs and to allow the transfer of the attack among aspect-level models.

$\mathbf{H}_{defenses}$: To determine the authenticity and integrity of text-input data allow to define an attack-independent defenses against adversarial examples to preserve correct model's results.

2.5 Objectives

2.5.1 General Objective

To define a model to generate task-oriented adversarial examples and to propose a preventive defense, both oriented to sentiment analysis at aspect-level, in order to impact on the correct models results according to the evaluation metric.

2.5.2 Specific Objectives

O1: To define a model to design task-oriented adversarial examples specially suited to sentiment analysis at aspect-level.

O2: To design and develop an adversarial attack applying task-oriented adversarial examples to evaluate their effectiveness to negatively impact on the accuracy of model's results.

O3: To evaluate the quality of task-oriented adversarial examples generated measuring the imperceptibility of modifications, the readability of the input as well as their effectiveness when they are transferred among models.

O4: To design and develop a preventive defense method based on the authenticity and integrity of text-inputs to guarantee correct results particularly of sentiment analysis models at aspect-level.

O5: To evaluate the effectiveness of the preventive defenses designed to maintain correct models' results in face of task-oriented adversarial examples.

2.6 Methodology

Figure 6 shown the work methodology of this research work, in following sections the involved activities are described at detail.

1. **Literature review**. A systematic literature review will be performed searching for content identification adversarial attacks and its application in text-based tasks. This review should critically emphasize the methods proposed to attack deep learning models for sentiment analysis task and the proposed defense mechanisms.

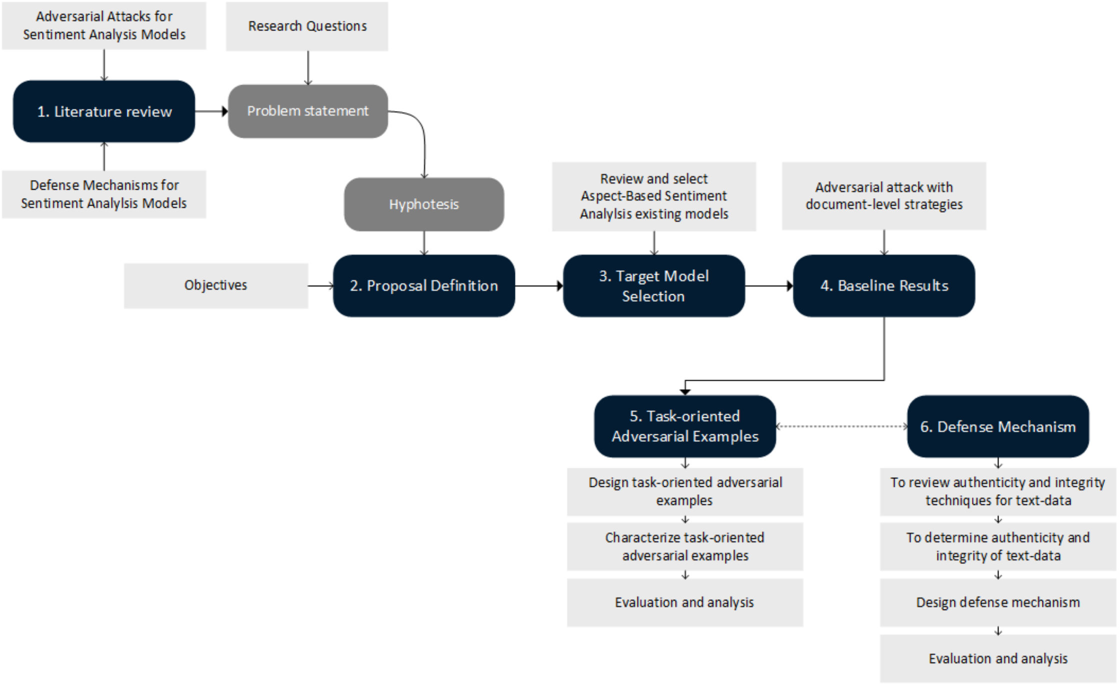

Adversarial Attacks for Sentiment Analysis Models
Research Questions
1. Literature review
Problem statement
Defense Mechanisms for Sentiment Analysis Models
Hyphotesis
Review and select Aspect-Based Sentiment Analylsis existing models
Adversarial attack with document-level strategies
Objectives
2. Proposal Definition
3. Target Model Selection
4. Baseline Results
5. Task-oriented Adversarial Examples
6. Defense Mechanism
Design task-oriented adversarial examples
Characterize task-oriented adversarial examples
Evaluation and analysis
To review authenticity and integrity techniques for text-data
To determine authenticity and integrity of text-data
Design defense mechanism
Evaluation and analysis

Figure 6: Overview of the research work methodology.

2. **Proposal definition**. Based on an extensive literature review and empirical knowledge, a problem statement and hypothesis will be proposed. This proposal should present novel strategies to approach the identified problem.

3. **Target model selection**. To experimental stages, a model for aspect-level sentiment analysis will be selected to be used as target model. Different promising open sources models for aspect-level will be reviewed; as adversarial examples exploits deep learning vulnerabilities, reviewed and selected models have to implement a DNNs architecture widely used in aspect-level analysis. Architectures as RNN, LSTM or CNN will be considered as well as transformers as BERT.

4. **Baseline results**. Based on literature review on stage 1, adversarial examples will be created by applying current strategies and techniques to modify text-inputs at document-level analysis to attack the target model selected on stage 3. As evaluation metrics, it will measure the accuracy of target model before an after an attack to observe the negative impact that baseline attack has on results. To evaluate the imperceptibility of modifications and the readability

of adversarial examples, it will measured the semantic similarity via cosine distance between x and x'. The obtained results will set the *baseline results* to evaluate, in following stages, the effectiveness of the attack and defense designed in this research work.

5. **Task-Oriented Adversarial Examples**. A model to generate adversarial examples oriented to aspect-level analysis will be defined. This model will consider aspect-level characteristics to determine opinion terms that must be modified to generate adversarial examples. This stage can be further divided in:

   (a) **Define task-oriented adversarial examples**. To generate task-oriented adversarial examples, it will be necessary to model the problem to reach that aspect-level and adversarial examples characteristics be combined. The main characteristic of analysis at aspect-level is the aspect-terms relation which indicates the opinion terms related to the evaluated aspect, for adversarial examples, it will be necessary to control the imperceptibility of modifications and the readability of the message.

   (b) **Design an task-oriented adversarial attack**. It will be designed an algorithm attack applying aspect-level adversarial examples model from stage 5a. This algorithm will identify the t_{a_i} by each mentioned aspect within an opinion to set the terms that should be modified to infringe the aspect-terms relation and, in this way, change the aspect's sentiment label. It will be necessary to consider that modifications maintain the $prox(a_i, (t+n)) \approx 1$ to look for the imperceptibility and readability properties.

   Since in this proposal we want to observe the transferability property, at first, the adversarial attack design will be oriented to a particular target model taking advantage of the knowledge of its technical characteristics. Later, task-oriented adversarial examples model will be evaluated on attacking different target models of which no technical details will be considered.

   (c) **Evaluation and analysis**. Adversarial attack designed on stage 5b will be evaluated on attacking a target model from stage 3. In this evaluation, the following activities are involved:

      i. **Evaluate the effectiveness of task-oriented adversarial examples**. A comparison of task-oriented attack results against baseline results from stage 4 will be carry out to observe the impact of the aspect-level adversarial examples on the accuracy of target's model.

   ii. **Evaluate the quality of task-oriented adversarial examples**. To evaluate the imperceptibility of modifications and the readability of the inputs based on semantic similarity of original inputs x and the generated adversarial examples x'.

(d) **Evaluate the transferability**. To evaluate the effectiveness of task-oriented adversarial examples model when it is transferred from one target model to other, for which, models selected on stage 3 will be used as target model to be initially attacked by baseline adversarial attack from stage 4 and later via task-oriented adversarial examples.

6. **Preventive defense mechanism against adversarial examples**. An attack-independent defense against adversarial examples will be defined. This defense will be focused on ensuring the integrity of text-inputs to guaranteeing they have not been modified by a third party. For this stage, it will be necessary to review existing techniques to determine the authenticity and integrity of text and thus to experiment on identifying modified text-inputs. Additionally, this stage could be divided in following activities:

   (a) **Design and develop a preventive defense mechanism**. To design a defense mechanism focused on determining the authenticity and integrity of text-inputs to identify modified inputs and discard them before feeding the model to avoid negative impact on training learning. To determine the authenticity and integrity of text, an input authentication algorithm have to be designed to works over target model to generate a secret key-space and validate that given the key-space and text-inputs they have not been modified.

   (b) **Evaluation and analysis**. As an end-to-end model, it will be evaluated the effectiveness of the preventive defense designed to maintain target model's accuracy against adversarial attacks generated from stages 4, 5b and 5d. This evaluation can be viewed as estimating (or modeling) as a preventive defense could work against different possibles modifications without have previous attack process knowledge.

2.7 Justification

Research efforts on design adversarial attacks for text-task emerges in recent years. Due to this recent growth, the volume and depth of contributions have been less than in other areas. Considering the potential of adversarial examples, there comes the need for identifying adversarial vulnerabilities and propose effective defenses for text-models (Swenor and Kalita, 2022). Until now, strategies for designing adversarial examples in text-task have been able to fool deep learning models and cause incorrect results. However, effective adversarial examples not only have to achieve a high rate in fooling DL models causing incorrect results, additionally they have to preserve the imperceptibility of modifications and maintain the readability of the input; the existing attacks methods have been struggling to preserve both properties. To generate effective adversarial examples, rather than focusing on exploring new combinations to make new modifications, it is necessary to propose new approaches as to consider the task properties to determine the terms to be modified. Considering task properties to generate task-oriented adversarial examples will permit to design modified inputs which exhibit imperceptibility and legibility. Additionally, basing defense methods on possible modifications performed to input data does not cover all possible modifications from an adversarial attack. Therefore, a design of a defense method focused on guaranteeing the authenticity and integrity of text-input data will make possible to propose mechanisms independent of a specific attack process and thus to maintain the correct model's results against adversarial examples.

2.8 Expected Contributions

The main contributions expected in this work are as follows:

C1: A model to design aspect-level adversarial examples.

C2: An adversarial attack particularly oriented to sentiment analysis models at aspect-level.

C3: A preventive adversarial defense for sentiment analysis models at aspect-level based on ensuring the authenticity and integrity of input data.

C4: An end-to-end model to defend sentiment analysis models for aspect-level in a preventive way against task-oriented adversarial examples.

2.9 Publications plan

According to the activities involved in the proposed methodology and work plan, the main publications to be developed are as follows:

- *Adversarial Adversarial Attacks: an Open Issue for Deep Learning Sentiment Analysis Models.* Sent to Neurocomputing Science Direct. Winter, 2022.
- *Aspect-Based Adversarial Examples for Sentiment Analysis Models.* Winter, 2023.
- *Determining authenticity and integrity of text-data.* Summer, 2024.
- *A preventive defense against adversarial examples for aspect-based sentiment analysis models.* Summer, 2025.
- *A preventive defense mechanism against aspect-based adversarial examples for sentiment analysis models.* Winter, 2025.

2.10 Work plan

Figure 7 presents the work plan to develop the activities to accomplish the objectives pursued in this research work.

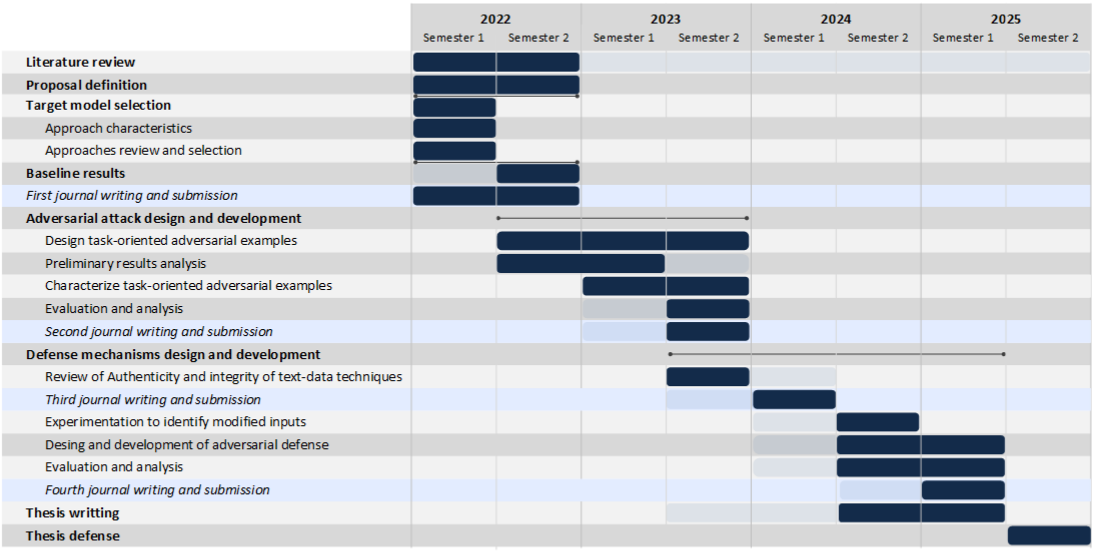


Figure 7: Work plan proposed to develop the activities involved in this PhD research work.

3 Background

In this section, it is included the preliminary knowledge related to sentiment analysis at aspect level and adversarial attacks on deep learning models for text-based tasks which cover current techniques and strategies to perform text-modifications.

3.1 Sentiment analysis

Nowadays, people and organizations use digital opinions to evaluate products or services and, in this way, make a decision. Usually, digital media have a large volume of opinions which are not always easy of filtering and analyzing. The average reader may have difficulty identifying reliable sites and accurately summarizing the information available. Likewise, individuals may have difficulty producing consistent results when the amount of information to be processed is extensive, moreover, human analysis is susceptible to personal biases, e.g., people often tend to pay more attention to opinions that coincide with their own preferences, discarding those that go against them. Because of these reasons, automatic opinion extraction and analysis systems are needed to overcome subjective biases and personal limitations to provide meaningful information (Liu and Zhang, 2012).

Sentiment analysis deals with the automatic extraction and analysis of opinions to identify emotions and attitudes and thus understand the sentiments expressed around products, services or topics of interest. (Liu and Zhang, 2012). Pang et al. (2002) define the sentiment analysis as *Computational processing of opinions, sentiments and subjectivity in texts.* This definition is one of the most popular and accepted. Nevertheless, Cambria et al. (2012) consider that this definition is general and ambiguous, therefore, they defined the sentiment analysis in a particular way: *Sentiment Analysis refers to the use of natural language processing, text analysis, and computational linguistics to identify and extract subjective information of the resources.* From a text-mining perspective, sentiment analysis is a task of automatically classifying documents based on the positive or negative connotation of the used language (Weiss et al., 2010). This task seeks to identify a user's attitudes regarding some topic of interest or determine the general contextual polarity (positive or negative) of a document.

Opinion elements. The term *entity* is used to indicate the object of which an opinion will be expressed. Typically, an opinion expresses the user's perception regarding certain components, characteristics or attributes of an entity and these components or characteristics related to an entity are called *aspects* (aspects are also called *target*, i.e. target opinion). Individuals or users who express an opinion, on their own or by an organization's name, are called *opinion sources.* The classification of opinions as positive, negative or neutral, is known as *determination of the polarity.* Polarity is determined based on the semantic orientation of the language used within an opinion, the *polarity* of an opinion indicates the type of sentiment expressed in it; the most common polarities are: positive, negative or neutral. For example, a particular brand of cell phone is an entity (e.g. *Samsung*). This entity has a set of aspects like battery, screen, quality, camera, and so on. Its users can express their acceptance or disagreement of its components; in other words, opinion sources express positive, negative or neutral sentiments about the entity's aspects.

3.1.1 Sentiment analysis systems

Prior to describing the concepts involved in a sentiment analysis system, the common terms within text-applications need to be mentioned:

- **Dataset** (N). A dataset is a set of texts that pertain to a specific topic. A dataset is, for example, each user opinion/comment/feedback about a product or service.
- **Document** (d). A dataset contains multiple texts, in the case of sentiment analysis a dataset contains multiple opinions. Each text in a dataset is identified as a document.
- **Word** (w). Words that are used in a document.
- **Term** (t). A term could be a n-gram (a word) or a n-gram (n words).
- **Vocabulary** (V). The distinct words or terms which form the documents are known as vocabulary.

3.1.2 Features extraction

The main objective of a sentiment analysis systems is to capture an effective set of features that allow to identify clearly and uniquely entities, aspects and polarities of sentiment (López Ramos and Arco García, 2019). Feature extraction is a process of identifying relevant features inside documents and representing them as numerical vectors to provide them to classification models. The different techniques for identifying the relevant features can be organized as follows:

- **Exploiting the frequency of terms occurring in the dataset.** For this purpose, techniques such as Bag of Words (BoW), Terms Frequency (TF) or Term frequency-Inverse document frequency (TF-IDF) can be used.

- **Analyzing syntactic and semantic relations, as well as dependencies and correlations between words/terms** (Hai et al., 2011; Zhai et al., 2011). Techniques such as *Part of Speech* (PoS), *cosine similarity* or *euclidean distance* are used.

- **Identifying opinion terms or opinion phrases to characterize polarities.** (Liu and Zhang, 2012). Opinion terms are words commonly used to express positive or negative sentiments. For example, *beautiful, wonderful, good, amazing* are positive opinion words and *bad, poor, terrible* are negative opinion words. Opinion words can be subjects or verbs that also indicate an opinion like *hate, like.* Not only exist individual opinion words but also opinion phrases exist. Opinion words and phrases are used in sentiment analysis to identify which terms present within an opinion correspond to positive or negative sentiments and then perform classification.

- **Designing customized strategies for feature extraction.** For example, the scoring function in (Dave et al., 2003) is based on giving scores according to probabilities for positive and negative opinion words.

3.1.3 Text representations

Text representation is the process of converting words into numbers for classification algorithms to understand and decode polarities' patterns (Ganegedara, 2018). The most used numerical representations in texts include:

- **Bag of Words (BoW)**. The Bag of Words model is used to represent documents ignoring word order. Under this model, a dictionary is created with the different terms (individuals or n-grams) present in the training set. Then, each opinion is represented as a bag containing the dictionary terms and the frequency of each one within the opinion (Sivic and Zisserman, 2008).

- **Terms and their frequency (Term frequency – Inverse document frequency, TF-IDF)**. The TF-IDF representation is a numerical measure that expresses the relevance of a word in a document (Beel et al., 2016). The frequency of a term t (individual or n-grams) in a document d, i.e. $tf(t,d)$, consists in determining the frequency of a term's occurrence in each document. When computing the tf, all terms are considered equally important; however, several terms can appear many times but have little importance. So, an inverse document frequency factor idf is incorporated to decrease the weight of terms that occur very frequently in documents and increase the weight of terms that rarely occur (Rajaraman and Ullman, 2011).

$$tf - idf(t,d) = tf(t,d) \cdot \log \frac{N}{df + 1} \tag{6}$$

- **One-hot encoding**. In one hot encoding, every word in a given text data is written in the form of V-dimension vectors, which are constituted only of 1s and 0s. All the word V-dimension vectors are combined to get a single document representation as a two-dimensional array.

- **Word Embeddings**. It allows words to be represented by a real value vector, often, of tens or hundreds of dimensions. Commonly, these vectors encode the meaning of the word and they are closer together in a vector space when have a similar meaning (Jurafsky and Martin, 2009).

3.1.4 Based-approaches

According to Liu and Zhang (2012), the proposed works can be grouped into three main approaches: i) lexicon-based, ii) learning-based, and iii) hybrid approach. In recent years, one of the approaches that have shown great success in tackling the task is deep learning.

- **Lexicon-based**. One of the most popular approach for analyzing opinions is the use of lexicons. Methods based on this technique use a set of *opinion words* as a tool to identify sentiment polarities. In this set, positive words are used to express approval or positive sentiments, meanwhile negative words represent dislike or displeasure sentiments (Liu and Zhang, 2012).

- **Learning-based**. Another approach to tackle the task focuses on building machine learning classifiers. The classifiers do not have prior knowledge of relevant features (such as opinion words) and instead they learn about features during training stage. As in other tasks, supervised, unsupervised or semi-supervised learning can be applied in sentiment analysis.
  - Supervised learning. The training of the classifiers is carried out using previously labeled data from which polarity labels indicate whether the document has a positive or negative connotation (Liu and Zhang, 2012).
  - Unsupervised learning. Unlike supervised learning, this learning can make use of unlabeled data to gain contextual information from extensive collections (Gonzalez, 2014).
  - Semi-supervised learning. It uses both labeled and unlabeled training data. Semi-supervised methods try to explore the structural information contained in unlabeled data to generate predictive models that perform better learning than those that only use labeled data (Gonzalez, 2014).
- **Hybrid**. Models of this type try to improve the performance of classifiers by combining two or more approaches. For example, some works experiment with the use of a supervised classifier fed by text representations based on lexicons (Severyn and Moschitti, 2015).
- **Deep learning-based**. One of the concepts that have been successful when applied to various domains of human knowledge (image processing, natural language processing, among others) is deep learning (Deng and Yu, 2014). Deep-learning models allow the characteristics of the input data to be learned at various abstraction layers and allow systems to learn most complex functions. In deep learning, artificial neural networks are used to facilitate the analysis of large volumes of information to identify characteristics of the study domain (López Ramos and Arco García, 2019).

3.1.5 Deep neural network models

A deep learning model is a set of machine learning algorithms that attempt to model high-level abstractions using deep neural network architectures that support multiple and iterative nonlinear transformations of data expressed in matrix or tensor form (Bengio et al., 2013). The models

Feed-Forward, Convolutional Neural Networks (CNN), and Recurrent/Recursive Neural Networks (RNN) and their variants have been the most implemented models for text tasks due to their natural ability to handle sequences and understand elements' relations. In sentiment analysis, the RNN type models are the most implemented. Particularly, the Long-Short Term Memory Networks (LSTM) and Convolutional Neural Networks (CNN) models since they can learn about sequences, locally and in the long term, preserving the most important and complex features that help the model to understand the complete relationships.

- **Recurrent Neural Network (RNN)**. RNN models can handle input sequences of variable length. RNNs create and process arbitrary memory sequences of input patterns and, unlike traditional methods for automatic sequence synthesis, RNNs models can process sequential and parallel information naturally and efficiently (Quintero and Garcia, 2018).

- **Long-Short Term Memory (LSTM)**. LSTMs models are a particular Recurrent Neural Network composed of units of the same type. Conventional RNN models can present problems in their training since the gradients tend to grow enormously or fade over time since gradient depends on present and past errors. The errors' accumulation could cause difficulties in memorizing dependencies in long texts. In LSTM, these problems are tackled by incorporating decisions about the information that is going to be stored and which one will be discarded (Graves, 2012).

- **Convolutional Neural Network (CNN)**. CNN models consist of multiple layers of convolutional filters of one or more dimensions. After each layer, a function is added to perform the non-linear causal mapping. At the beginning of the CNN, the feature extraction phase is composed of convolutional and downsampling neurons; as data is processed, its dimensionality decreases being the neurons in distant layers much less sensitive to data perturbations but, at the same time, they are activated by increasingly complex features[2]. At the end of the network, there are perceptron neurons to perform the final classification of extracted features.

[2] https:/ /www.juanbarrios.com/redes-neurales-convolucionales/

3.1.6 Evaluation metrics

For evaluating the performance of sentiment analysis systems, it is necessary to obtain a set of metrics to measure their effectiveness for classifying. The following metrics can be used to evaluate sentiment analysis systems' performance:

- **CA / Classification accuracy (AUC)**. Accuracy performance metric represents the proportion of correctly classified documents over the total number of processed documents.

$$AUC = \frac{True\ Positives + True\ Negatives}{All\ Samples} \tag{7}$$

- **Precision**. Precision measures the proportion of correct classifications that are actually correct; its value increases as the number of false positives decreases.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{8}$$

- **Recall / True Positive Rate (TP)**. Recall is the number of correct results divided by the number of results that should have been correct.

$$recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{9}$$

- **F1**. F1 measure is used to combine the measures of accuracy and recall into a single value, making it easier to compare the combined performance among various solutions.

$$recall = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{10}$$

3.1.7 Granularity levels

Sentiment analysis can be performed at different levels according to the text size or the objectives pursued. The granularity levels of sentiment analysis are divided as follows:

- **Document-level analysis**. The analysis of opinions at the document-level identifies the polarity of the sentiment expressed in a complete document considering it as a single unit of information (Pang et al., 2002). *Given a document, which evaluates an entity, it is determined the polarity of the sentiment expressed* (Liu and Zhang, 2012).

- **Sentence-level analysis**. Sentence-level opinion analysis consists in classifying the polarity of the expressed sentiment applied to individual sentences of a text (Riloff and Wiebe, 2003). In sentence-level analysis, two main tasks are involved: i) classification of subjectivity, in which it is determined whether the sentence is indeed an opinion, and ii) classification of the sentiment polarity of the sentence (Liu and Zhang, 2012).

- **Aspect-level analysis** (also known as Aspect-Based Sentiment Analysis - ABSA). In many cases, opinion analysis performed at the sentence or document level may not provide sensitive details of specific aspects. For example, a document with positive opinions concerning an entity does not mean that the author has positive opinions about all aspects of this entity. In order to identify those aspects regarding which the entities should pay more attention, the classification of sentiments is made at the aspect-level. Aspect-level analysis intends to identify the sentiment expressed about each aspect within an opinion individually (Hu and Liu, 2004; Liu et al., 2005). This analysis-level tries to overcome the limitation of analysis at document or sentence level when multiple aspects appear and each one is not evaluated independently (Henríquez et al., 2017).

3.1.8 Aspect-level analysis

The basic tasks of aspect-level sentiment analysis are: i) the extraction of the aspect(s) that have been evaluated in an opinion and ii) the classification of the sentiment, as positive, negative or neutral, on each aspect. On the other hand, in SemEval 2016 Evaluation forum (Pontiki et al., 2016), three important subtasks are established:

- *Aspect Category Detection; ACD.* This subtask relates to the identification and grouping of aspects into more general concepts such as food, decoration, cleanliness, etc. Specifically in Pontiki et al. (2016), this ACD seeks to identify the pair entity E and attribute A to which each aspect in an opinion refers. For example, for the opinion *"the spaghetti was tasteless"* the entity being discussed is $FOOD$ and its attribute $QUALITY$. The terms of a category do not necessarily appear specifically in an opinion but are inferred through the aspects present in it.

- *Opinion Target Expression; OTE.* This subtask aims at extracting the aspect terms. For example, in the opinion *"the spaghetti was tasteless"* the OTE on which the opinion is issued is the term *spaguetti*.

- *Sentiment Polarity; SP*. This subtask is in charge of assigning a sentiment (positive, negative or neutral) to the extracted aspects. Following the example of the first subtasks, for the opinion *""the spaghetti was tasteless"* the sentiment expressed is negative.

The different aspect-level subtasks can be developed independently, together (as a pipeline system) or take advantage of existing information. For example, to determine the sentiment polarity, aspect-level systems use the information of previously identified aspects.

3.2 Adversarial attacks for text-tasks

An adversarial attack consists of generating and inserting adversarial examples into input model to compromise its results. An adversarial example x' is a modified input created via a perturbation n of the input x to a DL model. The perturbation n is the minimal worst-case modification to input data which succeeds in confusing the model in its understanding and, as a consequence, in its classification (Zhang et al., 2020). A robust DL model should continue to classify the correct class y (according to the classification task) to x', while a victim model would have a high probability of wrong classification of x' (Zhang et al., 2020). x' can be formalized as follow:

$$\begin{gathered} x' = x + n,\ f(x) = y,\ x \in X \qquad (11) \\ f(x') \neq y \\ or f(x') = y',\ y' \neq y \end{gathered}$$

when n is the worst-case perturbation. The goal of the adversarial examples is deviating the label to incorrect one $f(x') \neq y$ or to an specific one $f(x') = y'$. Modifications to create adversarial examples should be as small as possible but capable of fooling DL models without changing human perception (refer to table 2).

3.2.1 Threat model

To identify the best criteria to design an adversarial attack, it is advisable to develop, test and analyze different modifications to determine which will be effective to fool the target model. In

Yuan et al. (2019), the crucial aspects to be taken into account when designing an adversarial attack are discussed, these criteria are described as follows:

- **Motivation**. Adversarial examples design is motivated by two objectives: attack or defense. Attack aims to examine the robustness of the target model, while defense uses the knowledge of adversarial examples to strengthen it.

- **Model Knowledge**. Adversarial examples can be designed under a black-box, white-box or grey-box scenarios (Zhang et al., 2020). The black-box attacks are performed when the details of target model are unknown and the adversarial examples are generally generated by accessing to test data or querying the target model and verifying an output change. In contrast, the white-box attack relies on the full knowledge of the technical details of target model. Lastly, the grey-box attack is a half-way point between black-box and white-box scenarios.

- **Target**. Adversarial examples can be generated to change the output prediction to: i) an specific class result (targeted) or ii) cause errors without any particular class (untargeted).

- **Granularity**. Refers to the detail-level at which modifications are performed. For text-applications, adversarial examples can be generated at character, term or sentence level.

  - *Character*. Modifications at this level can be summarized as insert, delete, swap or replace one or more characters seeking to preserve the terms' structure (Eger et al., 2019).

  - *Term*. This modification level include inserting, deleting, swapping or replacing a term (simple or n-gram) within a text attempting to preserve the semantics and syntax (Zang et al., 2019).

  - *Sentence*. Modifications at this level mainly perform the reordering of terms by paraphrasing the sentence and maintaining the original meaning of the message (Gan and Ng, 2019). Advanced methods aim at inserting text's fragments generated by using terms in data set.

  - *Multilevel*. Multilevel modifications combine changes at the character/term/phrase levels aiming to identify the optimal change to be performed (Liang et al., 2018; Vijayaraghavan and Roy, 2019).

3.2.2 Strategies

Modifications to generate adversarial examples can be performed by applying different strategies to modify certain input-terms or the complete input according to the granularity level. Text-based strategies to modify inputs include the following:

- **Concatenation**. This strategy consists of adding a sentence at the end of a text called as *distractor-text* to confuse the model without changing the semantics of the text (Jia and Liang, 2017).
- **Edit**. The attacks perform modifications to input data in two ways: i) *Synthetic*, the characters change order is made with *swaping*, *middle random* (random characters are exchanged except the first and the last one) and *fully random* (all the characters are randomly rearranged). ii) *Natural* in which the spelling errors in the original data are exploited. Advanced applications carry out modifications as: *Random Swap* by making an exchange of neighboring terms, *Stopword Dropout* by randomly removing empty words, *Paraphrasing* substituting terms by their paraphrase, *Grammar Errors* in which, for example, modifications are made via changing the conjugation of a verb, *Add Negation* and *Antonym strategy.*
- **Paraphrase-based**. Carefully produces a paraphrase of the original input with correct syntax and grammar.
- **Substitution**. This strategy attempts to reproduce the target model's operation in a local model to limit the requests to the victim model (Gil et al., 2019). In local model, potential adversarial examples that can confuse the target model are created and evaluated. If a potential adversarial example achieves to confuse the local model it is considered as adversarial example.

3.2.3 Modifications control

During adversarial attack's development, it is necessary to measure and control modifications in order to keep their size to a minimum, and after modifying the input, modifications size have to be measured to ensure that they are unnoticeable. Usually, the size of modifications is measured by the distance between the original data (or *clean data*) x and its adversarial example x'.

- **Gramar and Syntax measurement**. It is necessary to ensure correct grammar and syntax to make adversarial examples undetectable. Strategies 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 Euclidean distance) and similarity (such as cosine similarity).

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

  - *Jaccard similarity coefficient.* It is used to measure the similarity of finite sets using the intersection and union of the sets.

  - *Word Mover's Distance (WMD).* 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.* Levenshtein distance is a string metric for measuring difference between two sequences (minimum number of single-character edits) (Cormode and Muthukrishnan, 2007).

  - *Number of changes* is a simple way to measure edits.

3.2.4 Evaluation metrics

- **Success rate**. The success rate is the most direct and effective evaluation criteria (Zhang and Li, 2019). 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 to correct classifications. The robustness of DL models is related to the classification accuracy and how it is affected.

3.3 Adversarial defenses

The aim of defenses is to deal with modified inputs to identify and discard them to mitigate their negative impact on model's results. The existing defenses can be classified according their operations as follows:

1. **Investigate the differences between legitimate and adversarial texts**. For example, adversarial texts with character-level modifications show a notable difference from legitimate texts. Consequently, defense methods use an spell checker to detect this type of adversarial texts. However, modifications by substitution with synonyms are not detected.

2. **Improve DNN models to strengthen them against adversarial attacks**. This includes modifying the architectures to improve their safety and working with the training set with known parameters. (Madry et al., 2017) proposed DNN-model learning as a robustness optimization with min-max formulation, which is the composition of a non-concave internal maximization problem (attack) and a non-convex external minimization problem (defense).

3. **Adversarial training**. In this, the general idea is to introduce a set of known adversarial examples for neuronal network models learn from and thus avoid inconsistent results (classification errors) (Wang et al., 2021b; Wang and Wang, 2020). In (Szegedy et al., 2013), the authors proposed adversarial training. This strategy consists of training a neural network to classify both legitimate and adversarial texts correctly.

Unfortunately, previous defense have failed to deal with unknown adversarial attacks and rely on the knowledge process by which inputs were modified; but this is not practical since new attacks are constantly being designed. Papernot et al. (2016) proposed distillation as another defense against adversarial examples in which the objective is to use the softmax output (for example, the class probabilities when classification is made) of the original neural network to train a second model with the same structure.

4 Related work and state-of-the-art

Figure 8 depicts the systematic review methodology followed in this work. The literature review was performed on three primary data sources related to Computer Science and Data Security: IEEE, ScienceDirect, and SpringerLink. These data sources were considered as sources of information because they are the most credible databases since they are precisely and methodically peer-reviewed worldwide.

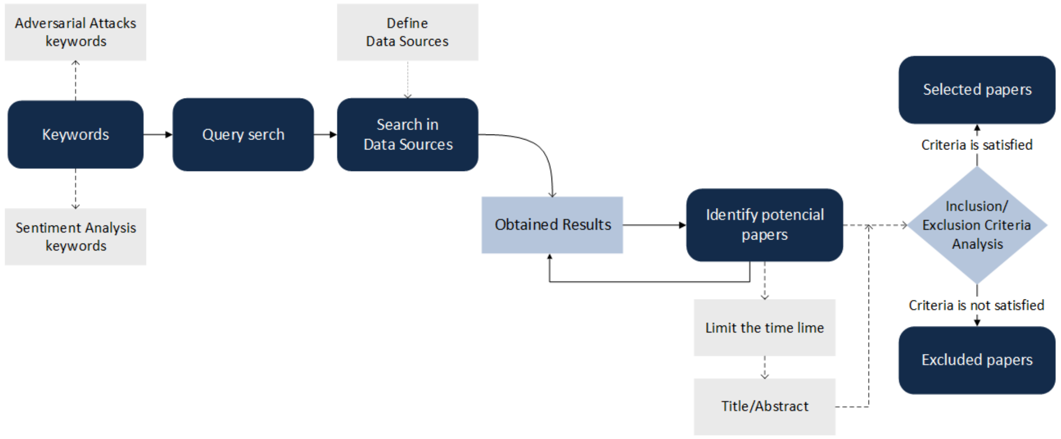


Figure 8: Systematic review methodology

4.1 Search query

First, keywords related to the central topic of this paper, Adversarial Attacks and Defenses, were defined, such as Adversarial, Attack, Examples, Defenses and Texts. Subsequently, keywords to delimit the research to sentiment analysis were added: Sentiment Analysis (SA) and Sentiment. Through the set of keywords defined (Adversarial, Attack, Examples, Defense, Text, OM, SA, Sentiment), the query string were integrated as: Adversarial Attacks in Sentiment Analysis, Adversarial Defenses in Sentiment Analysis. The review comprised the period from January to May 2022. Given that the research topic emerged in 2017, we limited the search to 6 years (2017 to 2022). The executed query string was: *("adversarial attack" or "adversarial texts") or ("adversarial defenses") and ("sentiment analysis" or sentiment).*

4.2 Articles selection methodology

The following methodology for the systematic review was implemented for the selection of the papers included in this work:

1. The keywords for the search string were defined according to the study case: Sentiment Analysis. The focus is on identifying papers that present an approach to design attacks or defenses, through adversarial examples, particularly for the case of study.
2. The sources of information are selected, and, through the search engine of these sources, the search for the query is defined in the advanced search section.
3. The search period is limited to 6 years (2017 to 2022).
4. The search is performed within all metadata (According to the specifications of each search database).
5. The search is delimited by type of publications to conferences, journals, papers, and magazines within the areas of Computer Science, Security, and Natural Language Processing.
6. The papers of the obtained results were reviewed to carry out an analysis considering inclusion and exclusion criteria.

4.3 Selection criteria

The paper selection process involves the evaluation of the results obtained from the information search and their filtering by the inclusion and exclusion criteria defined in this work. In the first instance, we exclude those papers oriented to text applications except for sentiment analysis. Also, we exclude those works that address different tasks, such as text classification in a generalized method. The inclusion and exclusion criteria for the selection of papers are formally defined as follows:

- The **inclusion criteria** consider 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 is given to papers published in the most recognized conferences in Natural Language Processing and Artificial Intelligence [3][4][5][6], although this is not a limitation.

- On the other hand, **exclusion criteria** are defined as:

1. The papers are not related to the area (text-based tasks).
2. The content and contributions of the papers are not related to the subject of research: adversarial attacks and their application in sentiment analysis.
3. Papers not related to sentiment analysis with deep learning models (or deep neural networks) approach..
4. The method presented in the papers is largely derived from other work.

4.4 Quality assessment and data extraction

We summarized the chosen research in the data extraction stage, so 232 articles were identified. At first, we checked the title and abstract of the articles and omitted those not aligned with the search purposes. Afterward, the inclusion criteria previously defined were followed, and 115 papers were selected. As a final step, the exclusion criteria were applied, and a total of 28 articles were selected as the initial research for the systematic review.

4.5 Adversarial attacks for sentiment analysis models

In table 3 the adversarial attacks for sentiment analysis models reviewed are summarized. According to the Threat Model characteristics, for the these works we indicated: model access, granularity, target and strategy applied. Additionally, we included the attacked DNN model, the considered metric to evaluate the effectiveness of the attack and finally the modification control applied. In following section, these works are described at detail.

[3] ACL: Annual Meeting of the Association for Computational Linguistics

[4] COLING: International Conference on Computational Linguistics

[5] EMNLP: Empirical Methods in Natural Language Processing

[6] IJCAI: International Joint Conference on Artificial Intelligence

Work	Model Knowledge	Granularity	Targeted	Strategy	DNN Model	Evaluation Metric	Modifications Control
Liang et al. (2018)	white-box	Character, Sentence	Targeted	Edit	CNN	Model robustness	-
Gong et al. (2018)	white-box	Character, Term	Untargeted	Hybrid	CNN	Model robustness	Word Mover's Distance
Li et al. (2018)	white-box	Character, Term	Untargeted	Edit	Character-level: LSTM Term-level: CNN	Sucess rate	Edit distance, Jaccard similarity, Euclidean distance and Semantic similarity
Tsai et al. (2019)	white-box	Term	Untargeted	Edit	CNN	Sucess rate	Perplexity, User evaluation, Semantic similarity
Alzantot et al. (2018)	black-box	Term	Untargeted	Edit	LSTM	Sucess rate	User evaluation
Ribeiro et al. (2018)	black-box	Term	Untargeted	Paraphrase-based	BiDAG, Visual7W, fastText	Sucess rate	Semantic similarity, User evaluation
Gao et al. (2018)	black-box	Character, Term	Untargeted	Hybrid	Character-level: CNN Term-level: LSTM	Model robustness	-
Jin et al. (2020)	black-box	Term, Sentence	Untargeted	Hybrid	CNN, LSM, BERT	Model robustness	Semantic similarity, User evaluation
Xu et al. (2021)	grey-box	Term	Untargeted	Edit	LSTM	Model robustness	Semantic similarity, Fluency

Table 3: Adversarial Attacks for Sentiment Analysis models.

4.5.1 Primary attack works

The principal objective of sentiment analysis models is to obtain an effective set of terms that uniquely identify different sentiments (positive, negative or neutral) which contribute to classify an opinion. Some authors refer to these terms as *valuable words* (Ma et al., 2018; Xiao and Zhou, 2020) since they have a crucial role in the final classification. Recent research seeks to determine with high precision those terms that contribute to the correct classification of input to using them to create adversarial examples (Wang et al., 2021a). In Liang et al. (2018) is presented a white-box adversarial attack denominated TextFool. TextFool is a targeted attack which uses the concept of FGSM (Fast Gradient Sign Method) to approximate the contribution of terms in a text to identify those that have a high impact on the input classification. In TextFool method, the adversarial examples are created by implementing three types of modifications at sentence-level: insert, modify (in which some characters are replaced) and delete. For its part, in Gao et al. (2018) the DeepWordBug method was proposed to generate small perturbations in texts in a black-box scenario. In DeepWordBug method, the *Replace-1 Score (R1S)*, *Temporal Head Score (THS)*, *Temporal Tail Score (TTS)* and *Combined Score (CS)* punctuation strategies are proposed to identify key terms that, if are modified, cause that classifiers make incorrect predictions. Character-level transformations are performed on the most relevant terms to minimize the edit distance of the perturbation from the original input.

The main difficulties in generating adversarial texts include: i) that input space is discrete, making difficult to accumulate small noises in the text-inputs and ii) measuring the quality of adversarial texts to preserve the modifications imperceptible. In Gong et al. (2018) in a white-box scenario, the discrete space is addressed by generating adversarial texts in the *embeddings* space against a CNN model, furthermore, the word mover's distance (WMD) is implemented to evaluate the similitude of the generated adversarial texts with original inputs. Li et al. (2018) presents a method called TextBugger in which is presented a perturbation constraint to evaluate the quality of adversarial texts generated in a white-box scenario by using different similarity measures: edit distance, Jaccard similarity coefficient, Euclidean distance and cosine similarity. For its part, (Tsai et al., 2019) propose a white-box method called *Global Search* in which simple modifications are made by adding spelling error noises with the intention to preserve the quality of the modifications under the idea that humans consider this type of errors as normal; additionally, this work propose a more sophisticated approach called *Greedy Search* in which the k nearest neighbors of each word in an opinion are chosen to be replaced and, to control the modifications, the perplexity is implemented to measure the degree of distortion (modification) of the generated adversarial examples.

On the other hand, among the challenges to be faced when generating adversarial texts are to preserve the correct semantic and syntax in order to maintain the legibility of the original input. To deal with this, (Alzantot et al., 2018) uses a population-based optimization algorithm to generate semantically and syntactically similar adversarial examples to try to fool sentiment analysis and textual entailment models. At first stage the main value words are identified and for each one, the nearest N synonyms neighbors which could replace it are searched into dataset.

Then, for selecting the correct synonyms to replace a word, the *Google 1 billion words language model* is used to discard those that are less frequent in the context of the text. Finally, from the remaining terms, it is selected the one that contributes more to the sentiment classification when it substitutes the original term. By another side, in Jin et al. (2020) the TextFooler method is proposed. This method uses two fundamental tasks of Natural Language Processing to generate adversarial examples: i) text classification and ii) textual entailment. According to the authors, using these tasks allow to preserve the semantic and grammatical content, and the correct human-classification. (Xu et al., 2021) present a gray-box adversarial attack and defense framework for sentiment classification. This work addresses issues of differentiability, label preservation, and input reconstruction for adversarial attack and defense in an unified framework.

4.6 Adversarial defenses for sentiment analysis models

This section presents the main defense works proposed for sentiment analysis models which have been a reference for the design of other defenses. In following section, main works are described.

4.6.1 Primary defense works

Proposed defenses against adversarial examples have mainly focused on implementing strategies as data augmentation and input preprocessing to try to identify modified entries and subsequently discard them (Zhou et al., 2019; Wang and Wang, 2020; Wang et al., 2021a).

The input preprocessing defenses require inserting an step between the input data and the given model to identify any possible modification. Based on this idea, in Pruthi et al. (2019) it is proposed a method for term validation and recognition before the input classification. This recognition method is based on the semi-character architecture of RNNs, introducing various feedback strategies for handling uncommon or unseen words. The method is trained to recognize words modified by random additions, swaps, or keyboard errors. Under this approach, the proposed defense achieves an error reduction of 32% in relative terms and 3.3% in absolute terms concerning to the conventional semi-character methods. Particularly, within the conclusions, authors argue that the proposed defense provides robustness to the classifier, improving both the adversary training and the standard spell checkers. For its part, Zhou et al. (2019) propose a DIScriminate Perturbations (DISP) defense mechanism for identifying and adjusting malicious modifications and blocking the attack. To identify adversary examples, the discriminator validates the probability that a term in the text is modified and 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 defense tries to block adversarial attacks without modifying models' structure or the training process. While, in Wang et al. (2019) authors propose 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. According to experimentation and observation, the authors conclude that SEM method can effectively defend against adversarial attacks based on synonym substitution.

Another popular defense based on identifying and discarding modified inputs is the method proposed by Wang et al. (2021a), in their proposal is presented a general defense mechanism called TextFirewall for different attacks under different strategies. Given an input text, TextFirewall identifies the modified text by evaluating the inconsistency between the output of the target model and the impact value calculated for the key terms in the text. Among the conclusions, the authors indicate that TextFirewall can be used as a tool without modifying the original model.

The design of adversarial example in text-based task has become popular in last years so, the volume and depth of contributions regarding defenses for text-applications as sentiment analysis has been less than for other tasks. Therefore, the exploration of effective attack strategies and defense mechanisms to ensure the correct functioning of the sentiment systems is clearly a necessity.

4.7 Discussion and open challenges

When designing textual adversarial examples, two critical and great challenges are present: preserving syntax and validating correct grammar and semantics. Additionally, there are challenges within text-applications that inherently need to be addressed for making the modifications imperceptible to humans but effective in confusing models. This could be one of the most challenging problems since a change within a text is easy to detect even if it was not intentional, such as orthographic errors. Another remaining challenge is related to ensuring the generality of the methods for the creation of adversarial examples, making them relatively easy to use in other models and preserving their effectiveness. In this section, according to reviewed works, we include the current challenges in designing adversarial examples.

- **Perceptibility**. While modifications on images are often imperceptible to human judgment, modifications on text are evident and readily identifiable. Invalid words and syntactic errors can be identified relatively easily using a grammar check process and thus be discarded. From a semantics preservation point of view, changing a word in a sentence could drastically change the semantics, and without additional process, the modified inputs could be identified and dismissed. In sentiment analysis applications, adversarial examples must be carefully designed not to change the expected output; otherwise, both correct and modified output changes infringe the purpose of generating adversarial examples. For an effective attack, approaches must be proposed not only to make the modifications imperceptible but also to preserve the correct grammar and semantics (Zhang et al., 2020). In Du et al. (2020), a white-box attack method against word-level CNN text classifier is presented. The approach uses Euclidean distance and cosine distance combined metrics 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 modification positions in generated examples more dispersed.

- **Transferability**. Transferability is an ideal property desired in the adversarial examples. This property reflects the generalization of attack methods by ensuring that the adversarial examples created for one model on a dataset can be used on another model or dataset while remaining effective (Zhang et al., 2020). Wiedeman and Wang (2022), propose that transferability between seemingly different models is due to a high linear correlation between the feature sets that different networks extract. In Yuan et al. (2020), a systematic investigation of factors that affect the transferability of adversarial examples for text classification models was explored. They contemplate factors such as network architecture, tokenization scheme, word embedding, and model capacity. Based on these studies, a genetic algorithm is proposed to find an ensemble of models that can be used to induce adversarial examples to fool different existing models.

- **Defenses attack-dependent**. Actual defenses against adversarial examples rely on the knowledge of generation process by which the model's inputs were modified, an approach that is not appropriate due to the increasing performance of the adversarial examples. To propose effective defenses, they have to be attack-independent which do not require the knowledge of generation process to identify modifications and discard adversarial examples, showing be more preventive rather than reactive.

- **New architectures**. Some architectures widely used in sentiment analysis have not been effectively attacked, for example, the generative models: Generative Adversarial Networks (GANs) and Variational Auto-Encoders (VAEs). Generative models require a great experience for model training, which may explain why they have not been effectively attacked. Recently, attention mechanisms have become a prevalent component in sequential models and have made it possible to improve the results obtained. However, there are no studies examining the functioning of these mechanisms and, thus, creating mechanisms for that they will be highly vulnerable.

5 Preliminary results

According to our work plan (refer to Fig. 7), in a first stage, our research work is focused on design particularized adversarial attacks oriented to cover task's characteristics.

5.1 Aspect-based adversarial examples

Following previous works, we pursue to design adversarial examples by modifying the value words (or most important terms) within an opinion. For aspect-level analysis, modifications should be made according to level's nature considering aspects evaluated, otherwise, irrelevant terms could be modified resulting in a loss of: i) modifications' imperceptibility and ii) the readability of the message. Given the nature of the aspect-based sentiment analysis, in order to generate adversarial examples, the terms to be modified have to be selected based on the evaluated aspects. For example, according to table 1, each aspect is related to specific terms by which is possible to determine the expressed user sentiment. On the basis of this main feature, the general adversarial examples formalization (refer to Eq. 1) have to be modified in such a way that they consider the aspect and opinion terms relation (aspect-term relation) and thus generate aspect-based adversarial examples.

We defined the aspect-based adversarial examples as follows: Given an opinion x consisting of n terms $x = \{t_1, t_2, ..., t_n\}$ with m different aspects mentioned $asp = \{asp_1, asp_2, ..., asp_m\}$. For each aspect asp_i there area different terms $t \in x$ particularly related to them $t_{asp_i} = \{t_{si}, t_{si} + l_i\}$ (l_i is the number or words in t_{asp_i} which express its ground truth user sentiment y_{asp_i} which should be understood and classified by the model M:

$$M(x, asp_i, t_{asp_i}) = y_{asp_i} \tag{12}$$

Therefore, to generate aspect-based adversarial examples, terms in t_{asp_i} have to be modified generating t'_{asp_i} and causing that M performs asp_i misclassification:

$$M(x', asp_i, t'_{asp_i})! = y_{asp_i} \tag{13}$$

At same time, x' should satisfy the following properties:

- Terms in t_{asp_i} can be uni-gram or n-gram words. To set t_{asp_i}, the proximity between aspect

asp_i and terms within x have to be computed; this proximity can be expressed as:

$$prox(a_i, t_i) = [0, 1] \tag{14}$$

A $prox(asp_i, t_i) \approx 0$ will be mean that t_i is not related to asp_i while $prox(a_i, t_i) \approx 1$ indicates a relation between t_i and a_i. So far, the $prox$ is calculated via cosine distance.

- To generate t'_{asp_i} each possible modification to $tint_{asp_i} = \{t_{si}, t_{si} + l_i\}$ should maintain the proximity to the original term, i.e $prox(t_i, t'_i) \approx 1$
- The modified input x' should be semantically similar to x. For this, $prox(t, t')$ is calculated via cosine distance between x and x'.

The idea behind this strategy lies on that by focusing on infringing the aspect-term relation, modifications to generate adversarial examples will be performed on the minimum necessary terms that effectively support aspect-sentiment and not across the complete opinion terms. This condition will contribute to perform the fewer modifications, maintaining the semantic similarity between the original inputs and the adversarial examples generated. To evaluate the effectiveness of our proposal, we designed an adversarial attack in a white-box scenario to generate an use aspect-based adversarial examples. In following sections the oriented-adversarial attack designed and the achieved results are presented.

5.2 Aspect-Based Adversarial Attack

Figure 9 illustrates our aspect-based adversarial attack designed (denominated as ABAA). Our attack was designed under a white-box scenario taking as a target model our previous approach: sentiment analysis using Specialized Aspect-Oriented Lexicons which proposes a term weighting scheme for aspect-level sentiment analysis. The approach takes as input a set of sentiment-oriented lexicons (according to dataset, refer to Table 4, for each category there is three lexicons, one by sentiment i.e. positive, neutral, negative) to model in a single vector each sentiment according to the average of the vectors of its terms and thus give a weight to each term within an opinion according to its semantic closeness concerning single vectors lexicons with this , terms pointing to sentiment in an opinion are highlighted allowing the sentiment classification. To evaluate the weighting scheme,

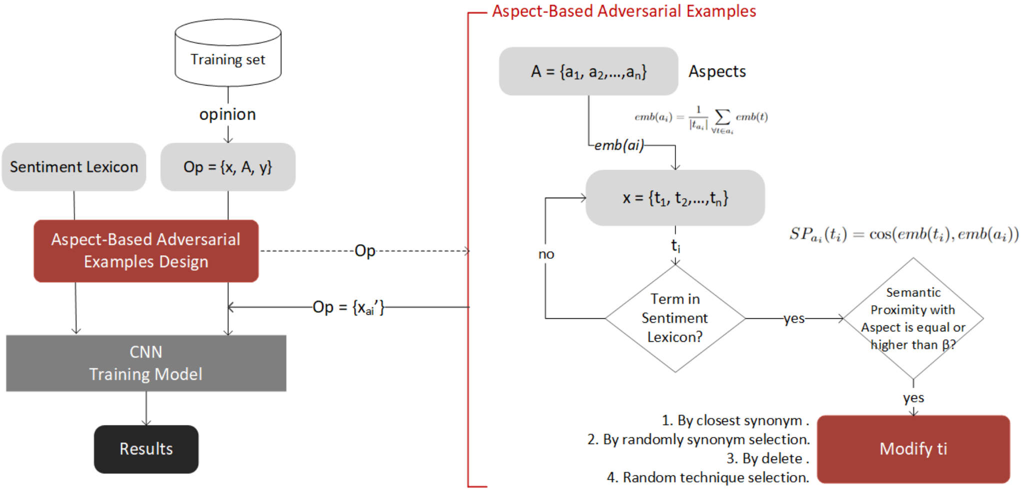

Training set
opinion
Sentiment Lexicon
Op = {x, A, y}
Aspect-Based Adversarial Examples Design
Op
Op = {x_{ai}'}
CNN Training Model
Results
Aspect-Based Adversarial Examples
A = {a_1, a_2,...,a_n}
Aspects
$emb(a_i) = \frac{1}{|t_{a_i}|} \sum_{\forall t \in a_i} emb(t)$
emb(ai)
x = {t_1, t_2,...,t_n}
t_i
no
$SP_{a_i}(t_i) = \cos(emb(t_i), emb(a_i))$
Term in Sentiment Lexicon?
yes
Semantic Proximity with Aspect is equal or higher than β?
yes
1. By closest synonym .
2. By randomly synonym selection.
3. By delete .
4. Random technique selection.
Modify ti

Figure 9: ABAA: Aspect-Based Adversarial Attack overview.

target model implements a CNN architecture using the SemEval[7] restaurant dataset which includes the information of aspects mentioned and their sentiment expressed by opinion. In a condensed way, figure 10 illustrates the target model methodology and table 4 describes the distribution of SemEval dataset.

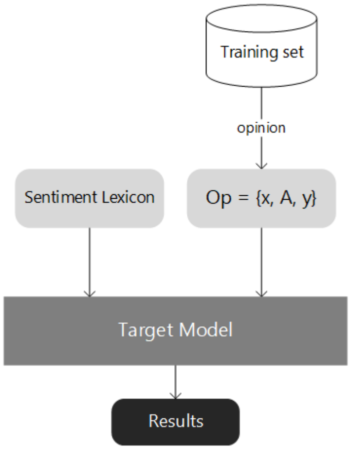


Figure 10: Target model methodology.

	instances	
Categories	**traing**	**test**
ambience#general	255	66
drinks#quality	47	22
drinks#style_options	32	12
drinks#prices	20	4
food#quality	849	313
food#style_options	137	55
food#prices	90	23
restaurant#general	422	142
restaurant#miscellaneous	98	33
restaurant#prices	80	21
service#general	449	155
location#general	28	13

Table 4: Distribution of SemEval 2016 dataset restaurant domain in English language.

[7] https://semeval.github.io/

Due to white-box attacks relies on full knowledge of target model's technical details to take advantage of this knowledge, to perform modifications and generated adversarial examples, we implemented the edit strategy to modify the terms in the sentiment-oriented lexicons since they are the most important terms for target model and they allow it to determine the sentiment polarity for each aspect within an opinion. Taking the sentiment-oriented lexicons and input data, adversarial examples are generated previous to training model and introduced to training dataset (refer to Fig. 9). To create aspect-based adversarial examples, term's modification were performed as follows:

1. Define an unique vector to represent the aspect evaluated referred as $emb(a_i)$. With $emb(t)$ indicating the embedding vector of each term t in aspect's term t_{a_i}, we computed $emb(a_i)$ as the average of the aspect's term vectors [8]:

$$emb(a_i) = \frac{1}{|t_{a_i}|} \sum_{\forall t \in a_i} emb(t) \tag{15}$$

2. For each term in sentiment lexicon, its semantic proximity SP is measured with respect to aspect a_i. The proximity is computed by the cosine similarity between the term and aspect's vector $emb(a_i)$:

$$SP_{a_i}(t_i) = \cos(emb(t_i), emb(a_i)) \tag{16}$$

3. To only modify the sentiment lexicon's terms most strongly associated with each aspect, we keep the terms whose semantic proximity is equal or above to β, considering $\beta = \{0.2, 0.3, ..., 0.6\}$. β is empirically defined considering that terms with semantic proximity close to 1 are terms that have the same direction as the aspect vector and, therefore, are strongly associated. Then, filtered terms are modified by applying a replace or delete technique as follows:

   - **Replace**. Replace in opinions the terms. For which, a list of synonyms by term is obtained and their semantic closeness is measured. Semantic closeness is defined as the cosine similarity between the original term and synonym. The synonym to replace the term can be selected by: i) the most semantic closely or ii) applying a random selection.
   - **Delete**. Filtered terms are delete in opinion.

[8] For represent terms and measuring semantic closeness, we use the pre-trained GloVe distributed embeddings on Twitter 200d.

Modifications techniques were tested one by one, and subsequently, a hybrid scenario is proved in which the modification technique to implement is randomly selected.

5.3 Baseline adversarial attack

As baseline results, we designed an adversarial attack applying an edit technique to modify the opinion terms which are in sentiment-oriented lexicons in opinions. In this attack, term's modification were performed as follows:

- **Replace**. In opinions, the terms contained in sentiment-oriented lexicons are replaced. By each term, a list of synonyms is obtained and their semantic closeness is measured. The synonym to replace a term lexicon is selected by: i) the most semantic closely or ii) applying a random selection.
- **Delete**. Sentiment-oriented lexicons terms contained in opinions are delete.

Term's modifications were tested one by one, and subsequently, a hybrid scenario was tested.

	BA
Target model	82.60 ± 0.46

Modification technique	**AA**	**SS**
Replace	74.48 ± 0.67	0.84
Random replace	79.28 ± 0.37	0.81
Delete**	74.50 ± 0.60	0.65
Hybrid	78.86 ± 0.47	0.73

Table 5: Baseline adversarial attack results by applied technique.

Table 5 presents the accuracy results obtained from target model before the attack (BA) as well as the achieved results when is applying the different modifications on training inputs, that is to say, accuracy after attack (AA). The results were calculated by executing ten times the target model; mean and ± std are shown. To evaluate the generated adversarial examples, the semantic similarity was measured by the cosine similarity between the original input x and modified input x'.

According to obtained results, we consider as baseline those results achieved by delete modification technique since it has the greatest impact on target model accuracy making it drops from 82.601% to 74.503% percent. In terms of attack success rate, table 6 presents the effectiveness of delete technique. After attack, the model resisted for 607 modified instances, leading a success rate of 9.806% (66/673) and an accuracy under attack (or after-attack accuracy) of 74.47% (607/815). Although deleting a terms means losing semantic, syntax and readability in the original inputs reaching a semantic similarity of only 0.65%, the baseline attack does not further mislead the target model.

Target model results	
Number of opinions	815
Number of predicted correctly	673
Number of predicted incorrectly	142
Accuracy	82.60
Attack results: Delete technique	
Number of reviews attacked	673
Number of succesful attack	66
Number of failed attack	607
Number of skipped attacks	142
Target model accuracy	82.60
Accuracy under attack	74.47
Attack succes rate	9.08%

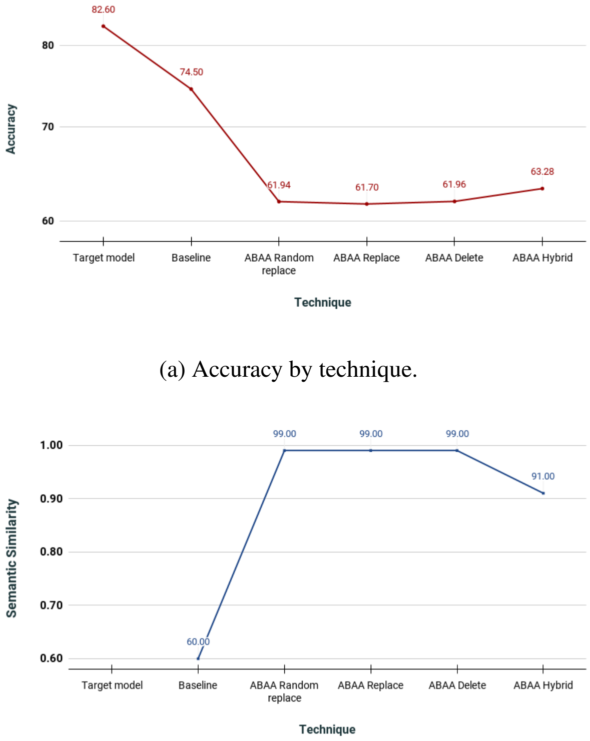
Table 6: Baseline attack success rate applying delete technique.

5.4 Experimental results

Table 7 presents the achieved results by aspect-based adversarial attack under the different modification techniques implemented. The results were calculated by executing ten times the target model to compute the accuracy; mean and ±std are shown. To evaluate the quality of generated adversarial examples, the semantic similarity was measured calculating the cosine similarity between the original input x and its modified input x'.

	ABAA Random Replace		**ABAA Replace**		**ABAA Delete**		**ABAA Hybrid**	
β	*AA*	*SS*	*AA*	*SS*	*AA*	*SS*	*AA*	*SS*
0.2	63.50 ± 0.74	0.86	61.88 ± 0.99*	0.93	63.28 ± 0.96	0.87	63.75 ± 0.85	0.90
0.3	63.50 ± 0.56	0.88	62.08 ± 0.90*	0.94	63.53 ± 0.69	0.89	63.58 ± 0.51	0.99
0.4	63.33 ± 0.68	0.91	62.23 ± 0.93*	0.95	63.32 ± 0.74	0.91	64.03 ± 0.59	0.91
0.5	62.66 ± 0.66	0.95	62.46 ± 0.47	0.97	62.31 ± 0.70*	0.96	**63.28 ± 0.65**	**0.92**
0.6	**61.94 ± 0.69**	**0.99**	**61.70 ± 0.37****	**0.99**	**61.96 ± 0.59**	**0.99**	63.56 ± 0.65	0.93
Target model *82.60 ± 0.47*								
Baseline *74.50 ± 0.60*								

Table 7: ABAA: Aspect-Based Adversarial attack results. In bold, the best results by applied technique are marked. Results marked with * indicate the best results according to β while results with ** indicate the best achieved results by ABAA attack.



(a) Accuracy by technique.

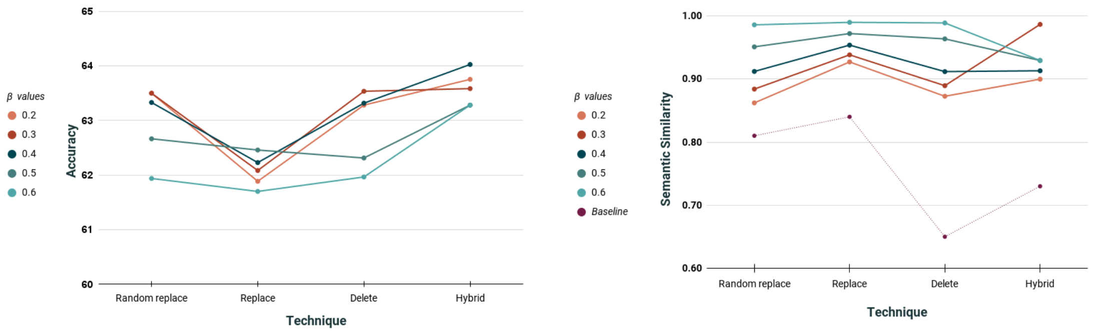
(b) Semantic similarity by technique.

Figure 11: Comparison ABAA results against target model accuracy and baseline attack results.

Figure 11 shows a comparison of ABAA results against target model accuracy and baseline attack results. When comparing the accuracy baseline results obtained and results using the aspect-based adversarial approach (refer to Fig. 11a), the usefulness of the proposed strategy is clearly appreciated. With ABAA, our best result is achieved with Replace technique filtering the terms to be modified with $\beta = 0.6$ (the biggest semantic proximity tested to filtering terms). Moreover, regarding the semantic similarity (refer to Fig. 11b), results show the effect that our approach provides to maintain the modifications as minimal as possible, achieving a semantic similarity of up to 0.99%. The ABAA results evidence the relevance of the proposed approach, since, it shows higher effectiveness to fool target model causing the accuracy from target model to drop a 20.90%; in addition, it outperforms the baseline results previously obtained. In terms of attack success rate, after ABAA attack, the model resisted for 503 modified instances, leading a success rate of 25.26% (170/673) and an accuracy under attack (or after-attack accuracy) of 61.70% (503/815).

5.4.1 Discussion

Figure 12 illustrates ABAA performance. On the one hand, figure 12a allows to evaluate the the effectiveness of each technique implemented. It is possible to appreciate that replace technique permits to obtain the best results by negatively impacting the target model's accuracy. Additionally, we can observe that applying a higher β to filter out the terms to be modified may be able to fool the target model with higher effectiveness.

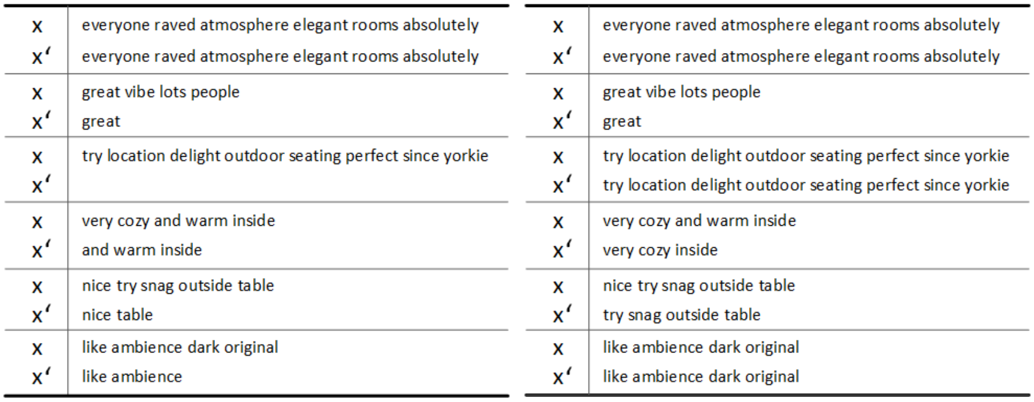


(a) Accuracy by technique according to β

(b) Semantic similarity by technique according to β.

Figure 12: ABAA achieved results.

On the other hand, figure 12b illustrates the positive effect that aspect-based adversarial examples have on preserving the semantic similarity between original inputs x and the adversarial examples generated x'. Additionally, in this figure we included the baseline semantic similarities previously observed; in contrast to baseline attack, it is evident that performing modification only for terms strongly related to aspects evaluated it is possible to maintain the input's readability due to modifications are minimal. For example, figure 13 presents adversarial examples generated from baseline attack and our ABAA proposals by delete technique. From both attacks we chose the same inputs that were modified; via this figure it is possible to observe the positive impact to maintain the input readability and the minimal size of term's modifications.



x	everyone raved atmosphere elegant rooms absolutely
x'	everyone raved atmosphere elegant rooms absolutely
x	great vibe lots people
x'	great
x	try location delight outdoor seating perfect since yorkie
x'	
x	very cozy and warm inside
x'	and warm inside
x	nice try snag outside table
x'	nice table
x	like ambience dark original
x'	like ambience

(a) Baseline adversarial examples by delete.

x	everyone raved atmosphere elegant rooms absolutely
x'	everyone raved atmosphere elegant rooms absolutely
x	great vibe lots people
x'	great
x	try location delight outdoor seating perfect since yorkie
x'	try location delight outdoor seating perfect since yorkie
x	very cozy and warm inside
x'	very cozy inside
x	nice try snag outside table
x'	try snag outside table
x	like ambience dark original
x'	like ambience dark original

(b) ABAA adversarial examples by delete with $\beta = 6$

Figure 13: Adversarial examples generated.

After evaluate the adversarial attack applying document-level techniques against our aspect-based adversarial attack, the principal remarks are:

- Document-level techniques fail to effectively fool the target model; even though the modifications create considerable changes and they impact on inputs readability, semantic and syntax.

- Since document-level modifications are not particularized to ABSA task, we observed that the techniques do not consider the relation of terms and aspects so the semantic connection throughout the text is not broken and, in a sense, there are no perturbations for the target model. Furthermore, when sentiment lexicon's terms are modified, the aspects' terms are not necessarily perturbed.

6 Final remarks

Unlike previous works our proposed model for generating aspect-based adversarial examples considers aspect term information to drive the modifications that must be performed to negatively impact on the robustness of the models. This latter characteristic makes that generated adversarial examples reduce the number of modifications which contribute to the imperceptibility of modifications and maintain the input readability, semantic and syntax. For the experimental stage, we determine aspect-term relation based on the semantic proximity of each term in an opinion with respect to the evaluated aspect to filtering the term that need to be modified. From the results obtained, it is possible to conclude that the aspect-based adversarial examples has a positive impact on fooling target model making that it accuracy drastically drops. Moreover, since terms to be modified are selected by semantic similarity, this brings the advantage of minimize the perceptibility of the modifications made. Although, it is necessary to continue to characterize the modifications in order to cover more task's characteristics such as the handling of negations or the control of terms that emphasize a certain sentiment expressed (e.g. very bad), this preliminary work shows promising results on achieving the objectives set in this research project.

As working directions, for the first semester of the second year of research, the main activity that we will focus on will be the characterization of aspect-based adversarial examples to later evaluate their their generality and transferability by carrying out adversarial attacks on different models and implementing different model knowledge scenarios (i.e. black-box and grey-box). Additionally, it will be necessary to use different contextual word embeddings and test it on new architectures such as those models which use BERT. Besides that, in a second stage of this work, it will be addressed the attack-dependence that actual adversarial defenses exhibit, for which a review of techniques to determine the authenticity and integrity of text data will be reviewed and based on acquired knowledge, to propose a preventive attack-independent to mitigate adversarial examples impact.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*, 2018.

Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. paper recommender systems: A literature survey. *International Journal on Digital Libraries*, 17(4):305–338, 2016.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Erik Cambria, Catherine Havasi, and Amir Hussain. Senticnet 2: A semantic and affective resource for opinion mining and sentiment analysis. In *Twenty-Fifth international FLAIRS conference*, 2012.

Graham Cormode and S Muthukrishnan. The string edit distance matching problem with moves. *ACM Transactions on Algorithms (TALG)*, 3(1):1–19, 2007.

Kushal Dave, Steve Lawrence, and David M Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528, 2003.

Li Deng and Dong Yu. Deep learning: methods and applications. *Foundations and trends in signal processing*, 7(3–4):197–387, 2014.

Xiaohu Du, Zibo Yi, Shasha Li, Jun Ma, Jie Yu, Yusong Tan, and Qinbo Wu. Generating more effective and imperceptible adversarial text examples for sentiment classification. In *International Conference on Artificial Intelligence and Security*, pages 422–433. Springer, 2020.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. Text processing like humans do: Visually attacking and shielding nlp systems. *arXiv preprint arXiv:1903.11508*, 2019.

Wee Chung Gan and Hwee Tou Ng. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1610. URL `https://aclanthology.org/P19-1610`.

Thushan Ganegedara. *Natural Language Processing with TensorFlow: Teach language to machines using Python's deep learning library.* Packt Publishing Ltd, 2018.

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

Yotam Gil, Yoav Chai, Or Gorodissky, and Jonathan Berant. White-to-black: Efficient distillation of black-box adversarial attacks. 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)*, pages 1373–1379, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1139. URL `https://aclanthology.org/N19-1139`.

Zhitao Gong, Wenlu Wang, Bo Li, Dawn Song, and Wei-Shinn Ku. Adversarial texts with gradient methods. *arXiv preprint arXiv:1801.07175*, 2018.

Andres Gonzalez. Conceptos básicos de machine learning. *Clever Data*, 2014.

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

Alex Graves. Long short-term memory. In *Supervised sequence labelling with recurrent neural networks*, pages 37–45. Springer, 2012.

Zhen Hai, Kuiyu Chang, and Jung-jae Kim. Implicit feature identification via co-occurrence association rule mining. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 393–404. Springer, 2011.

Carlos Henríquez, Ferran Pla, Lluís-F Hurtado, and Jaime Guzmán. Análisis de sentimientos a nivel de aspecto usando ontologías y aprendizaje automático. *Procesamiento del Lenguaje Natural*, pages 49–56, 2017.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, 2004.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*, 2017.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter 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*, volume 34, pages 8018–8025, 2020.

Daniel Jurafsky and James H Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson/Prentice Hall, 2009.

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

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. Deep text classification can be fooled. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4208–4215. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/585. URL `https://doi.org/10.24963/ijcai.2018/585`.

Bing Liu. *Web data mining: exploring hyperlinks, contents, and usage data*, volume 1. Springer, 2011.

Bing Liu and Lei Zhang. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer, 2012.

Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web*, pages 342–351, 2005.

Jiabao Liu, Qixiang Zhang, Kanghua Mo, Xiaoyu Xiang, Jin Li, Debin Cheng, Rui Gao, Beishui Liu, Kongyang Chen, and Guanjie Wei. An efficient adversarial example generation algorithm based on an accelerated gradient iterative fast gradient. *Computer Standards & Interfaces*, 82: 103612, 2022.

Dionis López Ramos and Leticia Arco García. Aprendizaje profundo para la extracción de aspectos en opiniones textuales. *Revista Cubana de Ciencias Informáticas*, 13(2):105–145, 2019.

Yukun Ma, Haiyun Peng, and Erik Cambria. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 135–147, 2017.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?' sentiment classification using machine learning techniques. *arXiv preprint cs/0205070*, 2002.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. Semeval-2016 task 5: Aspect based sentiment analysis. In *10th International Workshop on Semantic Evaluation (SemEval 2016)*, 2016.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, and Rada Mihalcea. Beneath the tip of the iceberg: Current challenges and new directions in sentiment analysis research. *arXiv preprint arXiv:2005.00357*, 2020.

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

Yisel Clavel Quintero and Leticia Arco Garcia. Estudio del análisis de sentimientos basado en aspectos. *"IV Conferencia Internacional en Ciencias Computacionales e Informáticas"*, 2018.

Anand Rajaraman and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2011.

Marco Tulio Ribeiro, Sameer Singh, and Carlos 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)*, pages 856–865, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1079. URL https://aclanthology.org/P18-1079.

Ellen Riloff and Janyce Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112, 2003.

Aliaksei Severyn and Alessandro Moschitti. Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 959–962, 2015.

Josef Sivic and Andrew Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 31(4):591–606, 2008.

Abigail Swenor and Jugal Kalita. Using random perturbations to mitigate adversarial attacks on sentiment analysis models. *arXiv preprint arXiv:2202.05758*, 2022.

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

Yi-Ting Tsai, Min-Chu Yang, and Han-Yu Chen. Adversarial attack on sentiment classification. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 233–240, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4824. URL https://aclanthology.org/W19-4824.

Prashanth Vijayaraghavan and Deb Roy. Generating black-box adversarial examples for text classifiers using a deep reinforced model. *arXiv preprint arXiv:1909.07873*, 2019.

Wenqi Wang, Run Wang, Jianpeng Ke, and Lina Wang. Textfirewall: Omni-defending against adversarial texts in sentiment classification. *IEEE Access*, 9:27467–27475, 2021a.

Xiaosen Wang, Hao Jin, and Kun He. Natural language adversarial attacks and defenses in word level. *arXiv preprint arXiv:1909.06723*, 2019.

Xiaosen Wang, Hao Jin, Yichen Yang, and Kun He. Natural language adversarial defense through synonym encoding. In *Conference on Uncertainty in Artificial Intelligence*, 2021b.

Zhaoyang Wang and Hongtao Wang. Defense of word-level adversarial attacks via random substitution encoding. In *International Conference on Knowledge Science, Engineering and Management*, pages 312–324. Springer, 2020.

Sholom M Weiss, Nitin Indurkhya, Tong Zhang, and Fred Damerau. *Text mining: predictive methods for analyzing unstructured information*. Springer Science & Business Media, 2010.

Christopher Wiedeman and Ge Wang. Disrupting adversarial transferability in deep neural networks. *Patterns*, 3(5):100472, 2022.

Yao Xiao and Guangyou Zhou. Syntactic edge-enhanced graph convolutional networks for aspect-level sentiment classification with interactive attention. *IEEE Access*, 8:157068–157080, 2020.

Ying Xu, Xu Zhong, Antonio Jimeno Yepes, and Jey Han Lau. Grey-box adversarial attack and defence for sentiment classification. *arXiv preprint arXiv:2103.11576*, 2021.

Liping Yuan, Xiaoqing Zheng, Yi Zhou, Cho-Jui Hsieh, and Kai-Wei Chang. On the transferability of adversarial attacksagainst neural text classifier. *arXiv preprint arXiv:2011.08558*, 2020.

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

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

Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. Clustering product features for opinion mining. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 347–354, 2011.

Jiliang Zhang and Chen Li. Adversarial examples: Opportunities and challenges. *IEEE transactions on neural networks and learning systems*, 31(7):2578–2593, 2019.

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

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