White-Box Adversarial Attacks Against Sentiment-Analysis Models using an Aspect-Based Approach

Monserrat Vázquez-Hernández[1], Ignacio Algredo-Badillo[1]*, Luis Alberto Morales-Rosales[2]* and Luis Villaseñor-Pineda[1]

[1] *Departamento de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Tonantzintla 72840, Puebla, México*
[2] *Facultad de Ingeniería Civil, CONAHCYT-Universidad Michoacana de San Nicolás de Hidalgo, Morelia 58000, Michoacán, México*

Abstract

Adversarial examples are deep learning model inputs that have been modified in their elements, by means of small and often imperceptible perturbations, which are able to confuse the models in the processing of the inputs and cause incorrect results. Among the issues still faced in adversarial example design for text applications, there is an absence of methods that model adversarial example design considering the characteristics of the task being addressed by DL models, as a consequence syntactically incorrect inputs are generated, impacting on the imperceptibility of modifications and the effective transfer of adversarial examples between models. In this work, we define a model for adversarial example generation, particularly oriented to aspect-based sentiment analysis on a white-box attack; this model considers aspect-based characteristics to drive the course of modifications. We evaluate our proposal model against adversarial examples generated for document-level analysis, demonstrating its effectiveness on impacting target model's results, making accuracy drops 20.90% and maintaining semantic similarities of adversarial examples in 99% concerning original inputs.

Keywords
adversarial attacks, vulnerabilities, aspect-based, sentiment analysis

1. Introduction

Sentiment analysis (SA), concerns the use of text analysis and machine-learning techniques for the automatic extraction and processing of users' opinions [1]. Sentiment-analysis systems are an important tool that provides summarized information concerning the experiences, positive or negative, that actual users have had about a product, service, or topic of interest.

Nowadays, sentiment analysis is used in many different areas to interpret users' opinions better. With this, organizations can propose improvements in products or services to enhance the experience of their users. For example, in the education field, the analysis of students' opinions collected from interactive learning environments, assisted collaborative learning, institutional digital media, or school administrative systems enables institutions to identify the sentiments expressed by students through their opinions and thus propose improvements to student experience [2]. Through students' comments, it is possible to track their learning behavior, progression, and experience and thus enhance the learning process according to students' needs. Understanding students' needs allows for transforming existing educational infrastructure to benefit students most.

CISETC 2023: International Congress on Education and Technology in Sciences 2023, December 04–06, 2023, Zacatecas, Mexico

mvazquez@inaoe.mx (M. Vázquez-Hernández); algredobadillo@inaoep.mx (I. Algredo-Badillo); lamorales@conacyt.mx (L. A. Morales-Rosales); villasen@inaoep.mx (L. Villaseñor-Pineda)
0000-0002-0877-7063 (M. Vázquez-Hernández); 0000-0002-4748-3500 (I. Algredo-Badillo); 0000-0003-1294-9128 (L. A. Morales-Rosales); 0000-0003-1294-9128 (L. Villaseñor-Pineda)

According to information needs, 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 [3], while at analysis sentence-level the objective is to analyze each sentence in a text to classify them [4]. Finally, the aspect-level analysis (or aspect-based sentiment analysis ABSA) seeks to independently determine the opinion expressed for each mentioned aspect within an opinion [5]. In many cases, the analysis at document or sentence-level does not provide specific details about particular aspects. For example, a negative document about an entity does not mean that the user has negative opinions about all aspects of this entity [6]; given this situation, it is necessary to work at a lower granularity, hence the interest and importance of aspect-level analysis. Table 1 illustrates the differences when performing the sentiment analysis at sentence and aspect levels.

Table 1
Aspect-based sentiment analysis. In bold are indicated the evaluated aspects

the **food** was *tasteless* but the **staff** was *pleasant*

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

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 recent years, the use of Deep Learning (DL) models to address aspect-level analysis has gained great popularity; through DL models, it is pursued to improve previous results and increase the confidence of its users, although this does not always turn out to be true. Several research works [7, 8, 9, 10, 11, 12] have demonstrated that DL models for applications using images or text can be effectively fooled 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 the original inputs to confuse the models on inputs' understanding and thus cause their incorrect classification (according to the classification task). Szegedy *et al.* [12] introduced adversarial examples when they studied the stability of state-of-the-art Deep Neural Networks (DNNs) for image classification in the face of modified inputs. Their work performed small pixel-level modifications to input data and observed that DNNs could be fooled by these modified inputs even if the human perception of data is not affected [13]. Based on the adversarial example idea, Jia and Liang in [11] 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 changing the original text, and they observed that DNNs text-models could also be fooled by adversarial examples. Since then, different works for the text-based task, such as [7, 8, 9, 10], have demonstrated that models can be fooled by making changes at character, term or sentence level by adding, deleting, substituting, or swapping text parts. Table 2 illustrates modified text-inputs by substituting a term with its synonym.

Previous works oriented to the Sentiment Analysis task [13, 6, 8] have proven to fool models effectively via adversarial examples. Although these works have impacted the accuracy results, they have mainly focused on addressing the task at the document-level and have not correctly dealt with aspect-based characteristics. Recently, Ekbal *et al.* in [15], proposed a method to generate adversarial examples oriented to aspect-based sentiment analysis, integrating specific

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

aspect-level characteristics to preserve opinion semantics. Their contribution relies on not modifying the terms of the aspect evaluated, so modifications to generate adversarial examples are made on the rest of the terms present in the opinion. Although the proposed attack contributes to achieving higher semantic similarity and grammatical correctness, it is assumed that only one aspect is evaluated within an opinion, which is not necessarily true (refer to Table 1). To generate effective adversarial examples, we consider that modifications to generate adversarial examples must be oriented to aspect-level in a particular way based on their characteristics. As table 1 shows, each aspect within the opinion directly relates to some opinion terms. On the aspect-level analysis, it is necessary to define this aspect-terms relation to determine the user's sentiment expressed correctly. Therefore, to perform input modifications and generate adversarial examples, the aspect-terms relation has to be identified, and changes have to be made to infringe the relation or change the aspect's sentiment. Suppose modifications are performed without considering aspect-terms relation; in that case, irrelevant terms can be modified without having the desired effect of adversarial examples, impacting the imperceptibility of modifications, the semantics and syntax of the texts, and the message's readability.

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

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

This work proposes a model for generating adversarial examples oriented to aspect-level analysis to deal with aspect-level characteristics correctly. Based on our proposed model, we define an adversarial attack to generate adversarial examples that are particularly oriented to aspect-based sentiment analysis models, identifying the terms directly related to the evaluated aspects and accordingly modifying them. We evaluate our adversarial attack against strategies previously applied for sentiment analysis which have proven to be effective in misleading DL models. Through achieved results, we show our attack's effectiveness since it outperformed the impact of the document-level adversarial attack by a 12.8% difference, maintaining a 99% semantic similarity between the original input and its adversarial example created. Moreover, our proposed attack shows its generality and transferability across contexts to be evaluated on different data sets, achieving context independence and maintaining the negative impact on model results. Summarizing, the main contributions of this work are:

- A model of aspect-based adversarial examples in which the modifications to be performed are conducted by considering aspect-level properties.
- A new strategy for deploying an adversarial attack, especially suited to aspect-level sentiment analysis to correctly deal with aspect-terms relation.
- An adversarial attack that achieves a higher semantic similarity and input's readability with fewer modifications.
- An adversarial attack that offers generality and transferability across different context data sets, maintaining the negative impact on aspect-level model's results and semantic similarity and input's readability.

2. Preliminaries

Before introducing the literature review and our proposal attack, this section includes preliminary knowledge related to adversarial attacks on deep learning models, particularly for text-based tasks, covering current techniques and strategies to perform text-modifications.

2.1 Adversarial examples

An adversarial attack consists of generating adversarial examples to fool the target model and negatively impact on its performance [13]. An adversarial example x' is an input generated by adding a perturbation n to an original input x of the target model, i.e. $x' = x + n$. A robust model should continue to classify the correct class y to x', while a victim model will have a high probability of incorrectly classifying x'. Zhang *et al*. in [13], presents a definition of adversarial examples and proposes the following formalization.

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

where n is the worst-case perturbation. The goal of the adversarial attack can be deviating the label to an incorrect one ($f(x') \neq y$) or specified one ($f(x') = y'$).

2.2 Threat model

In [16], the crucial aspects to be considered when designing an adversarial attack are discussed. These aspects are described as follows:

- **Model Knowledge.** Adversarial examples can be designed under a black-box, white-box or grey-box scenario. The black-box attacks are performed when the details of the target model are unknown. Generally, adversarial examples are generated by accessing test data or querying the target model and verifying an output change. In contrast, the white-box attack relies on knowledge of the technical details. Lastly, grey-box strategy 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) look for a specific class result (targeted) or ii) cause errors without any particular class (untargeted).
- **Granularity.** Refers to the detail level at which modifications are performed. In text-applications, adversarial examples can be generated at the character, term or sentence level, and the techniques to modify input data can be summarized as replace, delete, add or swap.
- **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.

To identify the best criteria to design an adversarial attack, it is necessary to develop, test and analyze different modifications at different levels to determine which will effectively fool the target model. For this reason, we experimented with designing different adversarial attacks according to *Model Knowledge*.

2.3 Strategies

To generate adversarial examples, we can apply different strategies to change specific terms of the input 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 a *distractor-text* to confuse the model without changing the semantics of the text [11].
- **Edit**. The attacks perform modifications to input data in two ways: i) *Synthetic*, the characters in a word or term are reordered via *swapping*, *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 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 [17]. Potential adversarial examples that could confuse the target model are created and evaluated in the local model. If a potential adversarial example achieves to confuse the local model, it is considered an adversarial example.

2.4 Modifications control

During adversarial example generation, that is to say when inputs are modified, it is necessary to measure and control modifications to keep them to a minimum size. Moreover, after input modification, it must measure the modifications' size to ensure their imperceptibility and the semantic of the text. Usually, adversarial examples are measured by the distance between the original data (or *clean data*) x and its adversarial example x'.

- **Grammar and Syntax measurement.** Ensuring correct grammar and syntax is necessary to make adversarial examples undetectable. Strategies such as perplexity measure, paraphrase control, and grammar and syntax checkers have been proposed to measure grammar and syntax.
- **Semantic-preserving measurement**. The semantic similarity/distance measurement is performed on word vectors using measures of distances (such as 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, for example: Jaccard similarity coefficient, Word Mover's Distance (WMD) or Perturbation ratio.

2.5 Evaluation metrics

For evaluating the performance of sentiment analysis systems, obtaining a set of metrics to measure their effectiveness is necessary. Therefore, we use the following metrics to evaluate sentiment analysis systems' performance:

- **Success rate.** The success rate is the most direct and effective evaluation criteria [18]. The attack success rate indicates the percentage of successful adversarial examples and the percentage of unsuccessfully attacked inputs. This measure provides insight into the susceptibility of a model to the designed adversarial examples.
- **Model Robustness**. Adversarial attacks are designed to affect the performance of models concerning the correct classifications. The robustness of DL models is related to the classification accuracy *Before-Attack-Accuracy* (BA) and how it is affected by adversarial examples *After-attack-accuracy* (AA).

3. Related work

Table 3 summarizes the adversarial attacks for sentiment analysis models reviewed. According to the threat model characteristics proposed in [16], we indicated for each work: the model knowledge, granularity, target (objective), and strategy applied. Additionally, we included the attacked DNN model, the considered metric to evaluate the effectiveness of the attack, and the modification control applied. In the following section, we describe these works in detail.

Table 3
Adversarial attacks for sentiment analysis models

Work	Model Knowledge	Granularity	Targeted	Strategy	DNN Model	Evaluation Metric	Modification Control
[14]	White-box	Character, Sentence	Targeted	Edit	CNN	Model robustness	-
[7]	White-box	Character, Term	Untargeted	Hybrid	CNN	Model robustness	Word Mover's Distance
[9]	White-box	Character, Term	Untargeted	Edit	LSTM, CNN	Success rate	Edit distance, Jaccard similarity, Euclidean distance and Semantic similarity
[8]	White-box	Term	Untargeted	Edit	CNN	Success rate	Perplexity, Semantic similarity
[10]	Black-box	Term	Untargeted	Edit	LSTM	Success rate	-
[19]	Black-box	Term	Untargeted	Paraphrase	BIDAG, Visual7W, fastText	Success rate	Semantic similarity
[20]	Black-box	Character, Term	Untargeted	Hybrid	CNN, LSTM	Model robustness	-
[21]	Black-box	Term, Sentence	Untargeted	Hybrid	CNN, LSTM, BERT	Model robustness	Semantic similarity
[22]	Grey-box	Term	Untargeted	Edit	LSTM	Model robustness	Semantic similarity, Fluency

3.1 Adversarial attacks for sentiment analysis models

The principal objective of sentiment analysis models is to obtain an effective set of terms that uniquely identify different sentiments (positive, negative, or neutral), contributing to classifying an opinion. Some authors refer to these terms as *valuable words* since they have a crucial role in the final classification [23, 24]. Recent research seeks to determine with high precision the terms that contribute to the correct input classification and use them to create adversarial examples [25]. Liang *et al*. [24] presented a white-box adversarial attack denominated TextFool. TextFool is a targeted attack that 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 the TextFool method, the adversarial examples are created by implementing three types of modifications at the sentence level: insert, modify (in which some characters are replaced), and delete. Gao *et al*. [20] proposed the DeepWordBug method to generate small text perturbations in a black-box scenario. In this 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 the classifier to make an incorrect prediction. 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 the input space is discrete, making it challenging to accumulate small noises in the text-inputs, and ii) measuring the quality of adversarial texts to preserve the modifications imperceptible. Gong *et al*. [7] proposed a white-box scenario, where 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. [9] presented a method called TextBugger, which is presented as a perturbation constraint to evaluate the quality of adversarial texts generated in a white-box environment using different similarity measures: edit distance, Jaccard similarity coefficient, Euclidean distance, and cosine similarity. Tsai *et al*. [8] proposed a white-box method called *Global Search*; they made simple modifications by adding spelling error noises to preserve the quality of the modifications under the idea that humans consider this type of errors as normal; additionally, a more sophisticated approach called *Greedy Search* is proposed, 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, one challenge to be faced when generating adversarial texts is preserving the correct semantics and syntax to maintain the original input's legibility. To deal with this, Alzantot *et al*. [10] used a population-based optimization algorithm to generate semantically and syntactically similar adversarial examples to try to fool sentiment analysis and textual entailment models. In the first stage, the main value words are identified, and of each word, the nearest *N* synonyms neighbors which could replace it are searched into the 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 substituting the original term. Jin *et al*. [21] proposed the TextFooler method. 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 allows the preservation of the semantic and grammatical content, as long as the correct human classification. Xu *et al*. [22] presented a gray-box adversarial attack and defense framework for sentiment classification, which addresses issues of differentiability, label preservation, and input reconstruction for adversarial attack and defense in a unified framework.

Up to now, most current works address different tasks by applying global strategies to modify inputs. This does not necessarily provide a correct solution since specific challenges in each task must be handled for a correct modification process. Although previous adversarial example attacks focusing on sentiment analysis have fooled models and reduced the accuracy of the results, these works have focused on addressing the sentiment analysis at the document-level and have not modeled the problem to deal with aspect-level characteristics. A recent attempt was made to attack an aspect-level classifier by Ekbal *et al*. [15] is proposed; this method integrates aspect-level characteristics to generate adversarial examples. In this method, given an evaluated aspect within an opinion, the terms that are part of the aspect are not modified but for all other terms in the opinion, it is intended to replace them by their synonym. The terms to be modified are selected according to their influence on the classification, to determine the influence of a term, it is masked with a special token and checked with the model to be attacked to see if it influences the classification. The proposed attack contributes to achieving higher semantic and syntax correctness; however, it is assumed that only one aspect will be evaluated, which is not necessarily true. By nature, different aspects can be included within an opinion, and different attitudes can be expressed for each of them. 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 which determine the positive or negative opinion by aspect (refer to Table 1); later for this identification, modifications to generate adversarial examples should be made on these terms.

We consider that an ideal adversarial-example design for aspect-level sentiment analysis should combine aspect-level and adversarial example characteristics to perform modifications on inputs and thus achieve task-oriented adversarial examples. First, to accomplish task-oriented adversarial examples, it will be necessary to correctly determine a set of terms that uniquely identify different sentiments (positive, negative, or neutral) contributing to classifying an opinion. Second, for each term, it will be necessary to establish the possible modifications *N* it should undergo, taking care of each one could be performed preserving the correct opinion semantics and syntax and successfully fool models.

4. Aspect-based adversarial examples

In contrast to the reviewed works, we aim to approach aspect-level sentiment analysis (or aspect-based. Hereafter, we will use aspect-based to identify the analysis at the aspect level). Given the nature of the aspect-based sentiment analysis, to generate adversarial examples, terms to modify need to be selected according to the evaluated aspects within an opinion. For example, according to Table 1, each mentioned aspect is related to specific terms by which it is possible to determine the expressed user's sentiment. Based on this main feature, the formalization of adversarial examples (refer to Eq. 1) must be modified to consider the aspect-terms relation and thus generate aspect-based adversarial examples.

We defined the aspect-based adversarial examples model as follows:

- Given an opinion x consisting of a set of terms T (and each t $\in T$ can be uni-gram or n-gram words) and a set A of different aspects mentioned within x. For each aspect $a \in A$ there is a term $t \in T$ particularly related to a_i which allows to understand and classify the expressed user' sentiment y_{ai}

$$\mathrm{M}(\mathrm{x}, a_i, \mathrm{t}_{ai}) = y_{ai} \quad (2)$$

- To identify the term $t \in T$ particularly related to a_i and set the relation aspect-term t_{ai}, the semantic proximity between aspect a_i and terms within x have to be computed; this proximity can be expressed as:

$$\mathrm{SP}(a_i, \mathrm{t}_i) = [0,1] \quad (3)$$

A $\mathrm{SP}(a_i, \mathrm{t}_i) \approx 0$ will be mean that t_i is not related to a_i while $\mathrm{SP}(a_i, \mathrm{t}_i) \approx 1$ indicates a relation between t_i and a_i .

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

$$\mathrm{M}(\mathrm{x}', a_i, t'_{ai}) \neq y_{ai} \quad (4)$$

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

- To generate t'_{ai} each possible modification to t_{ai} should maintain the semantic proximity to the original term, i.e. $\mathrm{SP}(t_i, t'_i) \approx 1$
- The modified input x' should be semantically similar to x. For this, the semantic similarity between x and x' is calculated and controlled.

The hypothesis behind our proposal lies on, by focusing on aspect-term relation, modifications to generate adversarial examples to negative impact on model's results, will be performed on the minimum necessary terms that effectively support aspect-sentiment which will contribute to perform the fewer modifications, maintaining modifications imperceptibility, inputs semantic and to allow the transfer of the attack among aspect-level models.

4.1 Adversarial attack

To evaluate the effectiveness of our proposal, we designed an adversarial attack in a white-box scenario to observe its performance. Figure 1 illustrates our aspect-based adversarial attack design (denominated as ABAA). The following sections describe the adversarial attack designed in this work and the achieved results.

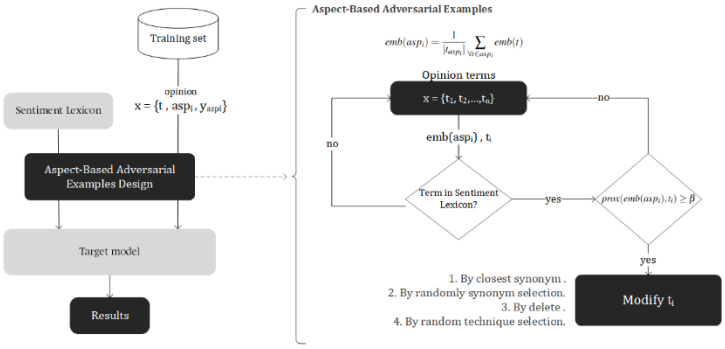


Figure 1: ABAA: Aspect-Based Adversarial Attack overview

4.2 Adversary's knowledge

Our adversarial attack takes as a target model our previous approach: Sentiment Analysis using Specialized Aspect-Oriented Lexicons [26], which proposes a term weighting scheme for aspect-based sentiment analysis. This approach takes as input a set of sentiment-oriented 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, the target model implements a CNN architecture using the SemEval[2] restaurant dataset. The restaurant dataset consists of two subsets: 1) a training set with 2,507 reviews and 2) a test set with 889 reviews. In both sets, the customer reviews include annotations identifying the aspects mentioned and its expressed sentiment polarity.

4.3 Aspect-Based adversarial examples design

Taking the sentiment-oriented lexicons and training set, adversarial examples are generated modifying reviews on test data, previous to training target model without affecting test set (refer to Fig. 1). To create aspect-based adversarial examples, the terms' modification was performed as follows:

Given an opinion x and one of the evaluated aspects within it ai:

1. Define the set of terms in aspect a_{terms} which includes the terms of the aspect a_i and the set of opinion terms x_{terms} which include the rest of the terms in opinion without the terms in a_{terms}. Let's consider the example "food is tasteless but the support staff was friendly". In this example, an evaluated aspect is support staff, thus the *set* $a_{terms} = (support, staff)$ and the set $x_{terms} = (food, is, tasteless, but, the, was, friendly)$.
2. Define a unique vector to represent the evaluated aspect. This unique vector is expressed as $\overrightarrow{emb}(a_i)$. We defined $\overrightarrow{emb}(a_i)$ as the average of the embedding[3] terms in a_{terms}:

$$\overrightarrow{emb}(a_i) = \frac{1}{|a_i|} \sum_{\forall\, t \in a_i} emb(t) \tag{5}$$

3. To only modify the terms associated with the aspect under evaluation, we identify terms in x_{terms} whose semantic proximity is equal or above to a threshold. The semantic

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

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

proximity SP is calculated by the cosine similarity between the embedding term t_i and $\overrightarrow{emb}(a_i)$:

$$SP(a_i, t_i) = \cos(\overrightarrow{emb}(a_i), emb(t_i)) \quad (6)$$
$$SP(a_i, t_i) \geq \beta$$

Then, filtered terms are modified by applying a replace or delete technique as follows:

- **Replace**. Replace in opinions the terms. For this, a list of synonyms by the 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 deleted in the opinion.

Modifications were tested one by one, subsequently, a hybrid scenario was tested. In the hybrid scenario, the modification to implement is randomly selected.

5. Experiments and results

Target model implements a CNN architecture using the SemEval restaurant dataset, which includes a training set and a test set with customer reviews with annotations identifying the aspects mentioned and the sentiment polarity of each aspect. We take advantage of the target model's technical details to drive modifications to generate adversarial examples. Specifically, we implemented the edit strategy to modify the terms in the sentiment-oriented lexicons since these are the most important terms for the target model, allowing it to determine the sentiment polarity for each aspect in an opinion. To filter terms to be modified, we consider $\beta = \{0.2, 0.3, 0.4, 0.5, 0.6\}$. We empirically defined β considering that terms with semantic proximity close to 1 have the same direction as the vector of the aspect term and, therefore, are strongly associated with the aspect evaluated and determine the user's sentiment expressed.

5.1 Reference adversarial attack

Prior to designing aspect-oriented adversarial examples, we designed a reference adversarial attack using the sentiment-oriented lexicons and training dataset. The reference attack consists of modifying opinion terms if these terms are in the sentiment-oriented lexicons without validating if they are related to the aspect being evaluated simulating an attack at document-level. The obtained results from this document-level reference attack, will serve to observe the potential and effectiveness of our proposal, which will allow us to determine the feasibility of conducting experiments in different scenarios (datasets, target models, modification technique, etc.) in order to compare it against current work that addresses the design of adversarial examples for aspect-level analysis.

In this reference attack, modifications were performed as follows:

- **Replace**. Taking advantage of sentiment lexicons used by the target model, all the terms in opinions are modified if they are included in the sentiment lexicons. A list of synonyms is obtained by each term, 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 lexicon terms contained in opinions are deleted.

As aspect-based adversarial attack, the modifications were tested one by one, and subsequently a hybrid scenario was evaluated.

5.2 Evaluation metrics

To measure the effectiveness of our proposal attack, we calculate i) Before-attack accuracy (BA) and After-attack-accuracy (AA), the before-attack-accuracy is calculated when any modification on training dataset is made, and After-attack-accuracy is calculated after opinions in training set are modified; ii) Attack success rate (SR), the percentage of adversarial examples that can successfully attack the target model; iii) Semantic similarity (SS): this is computed between the adversarial and actual sentence using the cosine similarity metric.

5.3 Results and analysis

Firstly, table 4 presents the results from the target model without any input modification and the results achieved when reference attack is applied. The results were calculated by executing ten times the target model; mean and standard deviation (± std) are shown. To evaluate the imperceptibility of generated adversarial examples, the semantic similarity was measured via the cosine similarity between the original input x and the modified input x'.

Table 4

Reference adversarial attack results by applied technique. It is presented BA: Before-attack-accuracy, AA: After-attack-accuracy and SS: Semantic similarity. In bold, the best results obtained are marked.

Technique	BA	AA	SS
Target model	82.60 ± 0.46	-	-
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

According to the obtained results in table 4, we consider as reference those results achieved by the delete technique since it has the greatest impact on target model accuracy, making it drop from 82.60% to 74.50% percent. In terms of attack success rate, the target model resisted for 607 modified instances, leading to a success rate of 9.806% (66/673) and accuracy after attack of 74.47% (607/815). Although deleting a term means losing semantics, syntax, and readability in the original inputs, the model reaches a semantic similarity of only 0.65% and the reference attack does not further mislead the target model.

Move on ABAA attack, Table 5 presents the accuracy after attack (AA) achieved by our proposed strategy under the different modification techniques implemented on the training dataset. The results are organized as follows: by each technique, we evaluate the selection of terms to be modified according to their semantic proximity to the evaluated aspect; as previously mentioned, this proximity is calculated by the cosine distance between the unique vector of the aspect's terms and the term under evaluation. For terms evaluation, we consider the $\beta = \{0.2, 0.3, 0.4, 0.5, 0.6\}$ to modify just terms with semantic proximity equal to or above β. The best results by the applied technique are marked in bold after applying the β threshold. The results marked with * indicate the best results according β to among the different techniques. Finally, the results with ** indicate the best-achieved results by the ABAA attack. The results were calculated by executing ten times the target model; mean and standard deviation (± std) are shown. The semantic similarity (SS) was measured by the cosine similarity between the original input x and modified input x' to evaluate the imperceptibility of modifications on generated adversarial examples.

Table 5

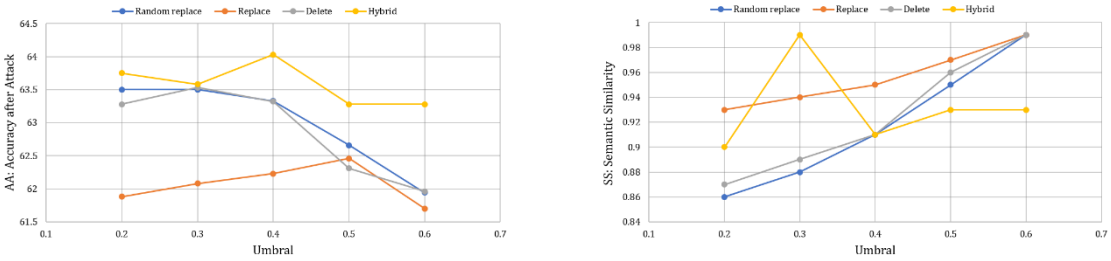
ABAA: Aspect-Based Adversarial attack results. It is presented AA: Accuracy-after-attack and SS: Semantic similarity. In bold, best results by applied technique are marked. Results marked with * indicate the best results according to while results with ** indicate the best achieved results

	ABAA Random replace		ABAA replace		ABAA delete		ABAA Hybrid	
	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
Reference attack 74.50 ± 0.60

The obtained results from ABAA evidence the relevance of the proposal since it shows higher effectiveness in fooling the target model, causing that accuracy target model drops by 20.90%. In terms of attack success rate, the ABAA attack outperforms reference results previously obtained. After the ABAA attack, the model resisted for 503 modified instances, leading to a success rate of 25.26% and an accuracy after attack of 61.70%.

Figure 2a illustrates the effect of each technique implemented on the accuracy model. It is possible to appreciate that the replace technique has a greater negative impact on the target model's results. In the same way, we can observe that applying a higher to filter out the terms to be modified, may be able to fool the target model with higher effectiveness. By other hand, figure 2b illustrates the positive effect of aspect-based adversarial examples on preserving the semantic similarity between original inputs x and the adversarial examples generated x'. In contrast to the reference attack, it is evident that modifying only terms related to aspects evaluated makes it possible to maintain the input's readability due to minimal modifications.



a) Accuracy by technique according to

b) Semantic similarity by technique according to

Figure 2: Comparison of obtained results from ABAA attack according to β

5.4 Discussion

Figure 3 permits to compare accuracy ABAA results against reference attack results. With ABAA, our best result is achieved with replace technique filtering the terms to be modified with β = 0.6 (refer to Fig. 2). Through this comparison, we can see the ABAA attack's effectiveness since it outperformed the impact of the document-level reference adversarial attack by a 12.81% difference. Via figure 4, it is possible to observe the positive effect that our proposal provides to maintain the modifications as minimal as possible, achieving a semantic similarity of up to 0.99% between the original input and its adversarial example created.

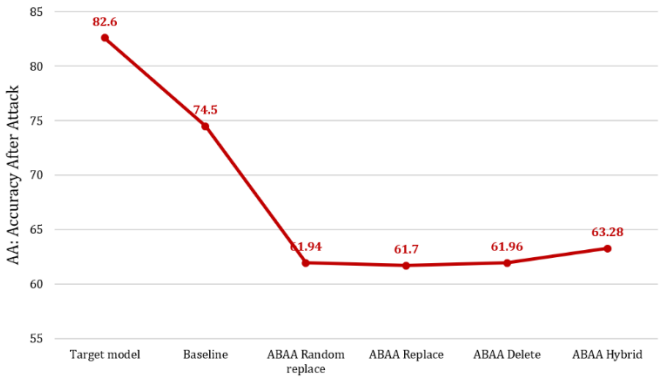


Figure 3: Comparison ABAA results against target model accuracy and reference attack results

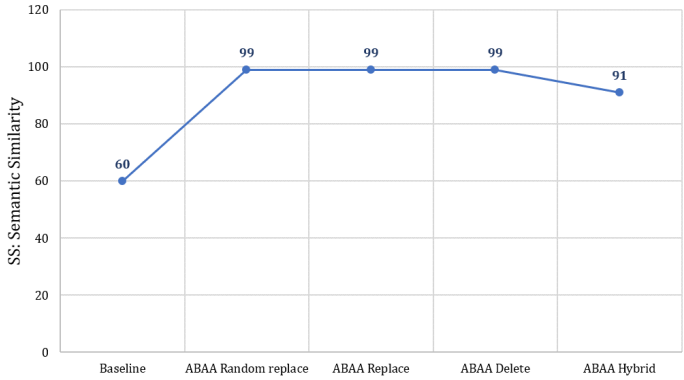


Figure 4: Comparison achieved ABAA semantic similarity by technique against reference semantic similarity

To illustrate the functionality of the filtering of terms according to the evaluated aspect, in table 6 are presented adversarial examples generated from reference attack and ABAA attack applying the delete technique. By means of these examples, it is possible to observe the positive impact of ABBA to maintain the input readability thanks to the minimal number of modified terms.

Table 6
Adversarial example generated via reference attack and ABAA attack.

Opinion	Adversarial examples from reference attack by delete technique	Adversarial example from ABAA attack by delete technique with $\beta = 0.6$
everyone raved atmosphere elegant rooms absolutely	atmosphere	everyone raved atmosphere elegant rooms absolutely
great vibe lots people	great	great vibe lots people
very cozy and warm inside	and warm inside	very cozy and warm inside
nice try snag outside table	nice table	nice try snag outside table
like ambience dark original	like ambiance	like ambience dark original

Observing the results obtained, and continuing with evaluation of our proposal's potential, to evaluate its generality and transferability, we performed another experiment for the same target model but using a different domain data set. In this case, we use the English laptop dataset from SemEval, as the restaurant dataset, the dataset consists of a training set and a test set, both containing customer reviews with annotations identifying the aspects mentioned and the sentiment polarity of each aspect. In table 7 are presented the obtained results in this evaluation. The results were calculated by executing ten times the target model; mean and standard deviation (± std) are shown.

Table 7

Aspect-Based Adversarial Attack results by applied technique on Laptop dataset. It is presented BA: Before-attack-accuracy, AA: After-attack-accuracy and SS: Semantic similarity. In bold, the best results obtained are marked

Technique	BA		AA	SS
Target model	77.48 ± 0.66		-	-
ABAA Replace	-	0.4	60.248 ± 3.11	0.786
ABAA Random replace	**-**	**0.5**	**60.011 ± 3.16**	**0.788**
ABAA Delete	-	0.4	60.110 ± 3.25	0.788
ABAA Hybrid	-	0.3	60.110 ± 3.88	0.776

As can be noticed, our attack significantly impacts the model's results. From this, we proved the generality and transferability across contexts of the aspect-based adversarial examples designed, proving its context independence since the ABAA attack maintains the same negative impact on the model's accuracy without any additional adjustment. The transferability of adversarial examples is an outstanding feature that adversarial strategies have to demonstrate when they are transferred from one model to another, maintaining their effectiveness. Until now, the lack of approaches that address tasks in a particularized way has prevented the effective transference of attacks among models, even when these attacks are carried out on the same task. Nevertheless, thanks to the results obtained during the evaluation of our proposal in different datasets, it is possible to observe the positive effect of particularizing the design of adversarial examples by considering the characteristics of the task; in our case, the aspect-level sentiment analysis can affect different domains without showing dependence on a specific context.

After evaluating our aspect-based adversarial attack against the reference attack and observing the performance of our proposal, the principal remarks are:

- Document-level techniques fail to fool the target model effectively, even though the modifications created considerable changes and impacted on input readability, semantics, and syntax (refer to Table 6). Through achieved results, we show our attack's effectiveness since it outperformed the impact of the document-level adversarial attack by a 12.81% difference.
- Since reference attack's modifications are not particularized to the 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 modifications for target model. Furthermore, due terms to be modified are not selected according to evaluated aspects, input terms (not related to aspects) are unnecessarily modified, impacting on modification's imperceptibility and, as consequence, on input readability. Making a comparison between adversarial examples from reference attack and ABAA attack (refer to Table 6), we observe that our proposal maintains a 99% semantic similarity between the original input and its adversarial example created.
- The obtained results from the evaluation of ABAA performance on a different dataset, show the generality and transferability of ABAA attack across different contexts, exhibiting context independence and maintaining relatively the same magnitude of negative impact on accuracy target model (refer to Table 7).

From the results obtained, we showed the relevance of the design of adversarial examples through modifications based on the task characteristics addressed with deep learning models. Besides, we demonstrated that the advanced modifications were designed to attack the target model, surpassing previous strategies effectively. Hence, we showed that our aspect-based adversarial examples effectively degrade the accuracy of the reference results obtained as well as in the semantics and syntax of the inputs preserving the fundamental characteristics of the adversary examples. In this sense, we carried out modifications as small as possible but capable of confusing the model.

The negative impact that the task-oriented adversarial examples have on the models compels us to continuously explore new vulnerabilities to propose defense mechanisms that allow them to be covered effectively and consequently guarantee the trust of the result obtained through the DL models. Therefore, we should analyze and propose further attack and defense methods since the models are susceptible to attacks. Particularly, the accuracy of the deep learning models can be decreased through adversarial examples, as we showed in this work. Different deep learning models implemented in various areas, such as sentiment analysis, have not yet completely solved their application problem, although improving a few percentage points or tenths of percentage points is an uphill task. Hence, considering research on attack and defense methods is critical since our attack presented decreased by over two tens of percentage points in the sentiment analysis task.

Thanks to the benefits that sentiment analysis models bring to the educational field, we consider that models should incorporate, from their design, defense mechanisms to prevent a future attack and mitigate negative consequences. We expect this work will motivate further research and development of new attacks and defense for educational sentiment analysis models.

6. Conclusions

The main contribution of this work is the formalization of aspect-based adversarial examples which considers the existing aspect-term relation to determine the terms to be modified. Unlike previous works, our proposed strategy for generating aspect-based adversarial examples considers aspect term information to drive the modifications that must be performed to negatively impact the models' accuracy. This latter characteristic ensures that adversarial examples maintain the input readability, semantics, and syntax obtaining a 99% semantic similarity between the original input and its adversarial example and making accuracy models drops 20.9%.

For the experimental stage, we determine aspect-term relation based on the semantic proximity of each term in an opinion concerning the evaluated aspect to filter the term that needs to be modified. From the results obtained, it is possible to conclude that the aspect-based adversarial examples have a positive impact on fooling the target model, making its accuracy drastically drops. Moreover, since terms to be modified are selected by semantic similarity, this minimizes the perceptibility of the modifications made. Besides, we evaluated the generality and transferability of our proposed aspect-based adversarial examples by evaluating them on different domain data sets, demonstrating context independence by maintaining a negative impact on model results. As working directions, we will evaluate our aspect-based adversarial examples on different target models with different adversary knowledge, different datasets, and architectures such as BERT or attention mechanisms, which will allow us to compare our proposal with current work that approaches adversarial examples generation for sentiment analysis models at aspect-level.

Acknowledgments

This work is supported by CONAHCYT/México scholarship 814461. Besides, it was founded by Catedras-CONAHCYT projects 882 and 613.

References

[1] B. Liu, Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Springer Berlin Heidelberg, 2011. URL: http://dx.doi.org/10.1007/978-3-642-19460-3. doi:10.1007/978-3- 642- 19460- 3.

[2] T. Shaik, X. Tao, Y. Li, C. Dann, J. McDonald, P. Redmond, L. Galligan, A review of the trends and challenges in adopting natural language processing methods for education feedback

analysis, IEEE Access 10 (2022) 56720–56739. URL: http://dx.doi.org/10.1109/ACCESS.2022.3177752.doi:10.1109/access.2022.3177752.
[3] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? sentiment classification using machine learning techniques, in: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), Association for Computational Linguistics, 2002, pp. 79–86. URL: https://aclanthology.org/W02-1011. doi:10.3115/1118693.1118704
[4] E. Riloff, J. Wiebe, Learning extraction patterns for subjective expressions, in: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, 2003, pp. 105–112. URL: https://aclanthology.org/W03-1014.
[5] S. Poria, D. Hazarika, N. Majumder, R. Mihalcea, Beneath the tip of the iceberg: Current challenges and new directions in sentiment analysis research, IEEE Transactions on Affective Computing 14 (2023) 108–132. URL: https://doi.org/10.1109/taffc.2020.3038167. doi: 10.1109/taffc.2020.3038167
[6] B. Liu, L. Zhang, A survey of opinion mining and sentiment analysis, in: Mining text data, Springer, 2012, pp. 415–463. doi: 10.1007/978-1-4614-3223-4_13
[7] Z. Gong, W. Wang, B. Li, D. Song, W.-S. Ku, Adversarial texts with gradient methods (2018). URL: https://arxiv.org/abs/1801.07175. doi:10.48550/ARXIV.1801.07175.
[8] Y.-T. Tsai, M.-C. Yang, H.-Y. Chen, Adversarial attack on sentiment classification, in: Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Association for Computational Linguistics, Florence, Italy, 2019, pp. 233–240. URL: https://aclanthology.org/W19-4824. doi: 10.18653/v1/W19- 4824.
[9] J. Li, S. Ji, T. Du, B. Li, T. Wang, TextBugger: Generating adversarial text against real-world applications, in: Proceedings 2019 Network and Distributed System Security Symposium, Internet Society, 2019. URL: https://doi.org/10.14722/ndss.2019.23138. doi: 10.14722/ndss.2019.23138.
[10] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, K.-W. Chang, Generating natural language adversarial examples, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2018. URL: https://doi.org/10.18653/v1/d18-1316. doi:10.18653/v1/d18- 1316.
[11] R. Jia, P. Liang, Adversarial examples for evaluating reading comprehension systems, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2017. URL: https://doi.org/10.18653/v1/d17-1215. doi:10.18653/v1/d17- 1215.
[12] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199 (2013).
[13] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, C. Li, Adversarial attacks on deep-learning models in natural language processing: A survey, ACM Transactions on Intelligent Systems and Technology 11 (2020) 1–41. URL: http://dx.doi.org/10.1145/3374217. doi:10.1145/3374217.
[14] B. Liang, H. Li, M. Su, P. Bian, X. Li, W. Shi, Deep text classification can be fooled, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 4208–4215. URL: https://doi.org/10.24963/ijcai.2018/585. doi:10.24963/ijcai.2018/585.
[15] Mamta, A. Ekbal, Adversarial sample generation for aspect based sentiment classification, in: Y. He, H. Ji, S. Li, Y. Liu, C.-H. Chang (Eds.), Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022, Association for Computational Linguistics, Online only, 2022, pp. 478–492. URL: https://aclanthology.org/2022.findings-aacl.44.
[16] X. Yuan, P. He, Q. Zhu, X. Li, Adversarial examples: Attacks and defenses for deep learning, IEEE Transactions on Neural Networks and Learning Systems 30 (2019) 2805–2824. URL: http://dx.doi.org/10.1109/TNNLS.2018.2886017. doi:10.1109/tnnls.2018.2886017.
[17] Y. Gil, Y. Chai, O. Gorodissky, J. 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), Association for Computational Linguistics, Minneapolis, Minnesota,

2019, pp. 1373–1379. URL: https://aclanthology.org/N19-1139. doi:10.18653/ v1/N19-1139.
[18] J. Zhang, C. Li, Adversarial examples: Opportunities and challenges, IEEE Transactions on Neural Networks and Learning Systems (2019) 1–16. URL: http://dx.doi.org/10.1109/TNNLS.2019.2933524. doi:10.1109/tnnls.2019.2933524.
[19] M. T. Ribeiro, S. Singh, C. Guestrin, Semantically equivalent adversarial rules for debugging NLP models, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 856–865. URL: https://aclanthology.org/P18-1079. doi:10.18653/v1/P18- 1079.
[20] J. Gao, J. Lanchantin, M. L. Soffa, Y. Qi, Black-box generation of adversarial text sequences to evade deep learning classifiers, in: 2018 IEEE Security and Privacy Workshops (SPW), IEEE, 2018. URL: http://dx.doi.org/10.1109/SPW.2018.00016. doi:10.1109/spw.2018.00016
[21] D. Jin, Z. Jin, J. T. Zhou, P. Szolovits, Is BERT really robust? a strong baseline for natural language attack on text classification and entailment, Proceedings of the AAAI Conference on Artificial Intelligence 34 (2020) 8018–8025. URL: https://doi.org/10.1609/aaai.v34i05.6311. doi:10.1609/aaai.v34i05.6311.
[22] Y. Xu, X. Zhong, A. Jimeno Yepes, J. H. Lau, Grey-box adversarial attack and defence for sentiment classification, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2021. URL: http://dx.doi.org/10.18653/v1/2021. naacl-main.321. doi:10.18653/v1/2021.naacl-main.321.
[23] Y. Ma, H. Peng, E. Cambria, Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm, volume 32, Association for the Advancement of Artificial Intelligence (AAAI), 2018. URL: http://dx.doi.org/10.1609/aaai.v32i1.12048. doi:10.1609/aaai.v32i1.12048.
[24] Y. Xiao, G. Zhou, Syntactic edge-enhanced graph convolutional networks for aspect-level sentiment classification with interactive attention, IEEE Access 8 (2020) 157068–157080. URL: http://dx.doi.org/10.1109/ACCESS.2020.3019277. doi:10.1109/access.2020.3019277.
[25] W. Wang, R. Wang, J. Ke, L. Wang, Textfirewall: Omni-defending against adversarial texts in sentiment classification, IEEE Access 9 (2021) 27467–27475. URL: http://dx.doi.org/10.109/ACCESS.2021.3058278. doi:10.1109/access.2021.3058278.
[26] M. Vázquez-Hernández, L. Villaseñor-Pineda, M. Montes-y Gómez, A semantic-proximity term-weighting scheme for aspect category detection, Procesamiento del Lenguaje Natural (2022) 117–127. URL: https://doi.org/10.26342/2022-69-10. doi:10.26342/2022- 69- 10.