

# Arabic Synonym BERT-based Adversarial Examples for Text Classification

Norah Alshahrani      Saied Alshahrani      Esma Wali      Jeanna Matthews  
Department of Computer Science, Clarkson University, Potsdam, New York, USA  
{norah, saied, walie, jnm}@clarkson.edu

## Abstract

Text classification systems have been proven vulnerable to adversarial text examples, modified versions of the original text examples that are often unnoticed by human eyes, yet can force text classification models to alter their classification. Often, research works quantifying the impact of adversarial text attacks have been applied only to models trained in English. In this paper, we introduce the *first* word-level study of adversarial attacks in Arabic. Specifically, we use a synonym (word-level) attack using a Masked Language Modeling (MLM) task with a BERT model in a black-box setting to assess the robustness of the state-of-the-art text classification models to adversarial attacks in Arabic. To evaluate the grammatical and semantic similarities of the newly produced adversarial examples using our synonym BERT-based attack, we invite four human evaluators to assess and compare the produced adversarial examples with their original examples. We also study the transferability of these newly produced Arabic adversarial examples to various models and investigate the effectiveness of defense mechanisms against these adversarial examples on the BERT models. We find that fine-tuned BERT models were more susceptible to our synonym attacks than the other Deep Neural Networks (DNN) models like WordCNN and WordLSTM we trained. We also find that fine-tuned BERT models were more susceptible to transferred attacks. We, lastly, find that fine-tuned BERT models successfully regain at least 2% in accuracy after applying adversarial training as an initial defense mechanism.

## 1 Introduction

Machine Learning models, in general, are vulnerable to adversarial attacks, which are small, crafted perturbations done usually by altering the original input of these models in order to change their classification (Huang et al., 2011; Corona et al., 2013; Kurakin et al., 2016; Kumar et al., 2017; Pitropakis et al., 2019; Wiyatno et al., 2019; Mello, 2020).

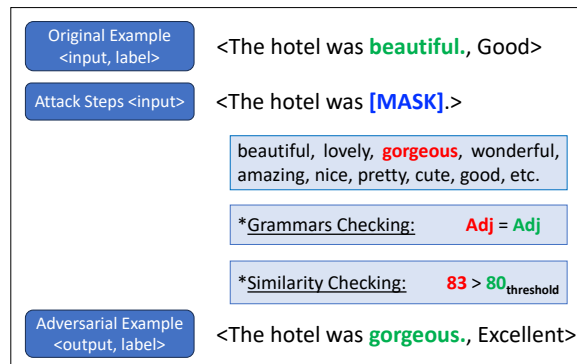


Figure 1: A diagram illustrates our attack steps for generating synonym adversarial attacks using an MLM task with BERT. The attack first predicts synonym tokens and then checks their grammar and semantic similarities. Once a predicated token satisfies the grammar and semantic checkings, we have an attack candidate example.

Research on adversarial attacks is often in the domain of image classification systems (Szegedy et al., 2014; Kurakin et al., 2017a,b; Papernot et al., 2017; Kos et al., 2018; Arnab et al., 2018) or speech recognition systems (Hannun et al., 2014; Kereliuk et al., 2015; Gong and Poellabauer, 2017; Carlini and Wagner, 2018; Samizade et al., 2020; Želasko et al., 2021). For example, in the domain of image classification, the images are the original inputs, and the attackers could negatively affect the performance of these systems by introducing small perturbations to the input images (Szegedy et al., 2014; Kurakin et al., 2017a,b; Papernot et al., 2017; Liang et al., 2018). Examining adversarial attacks in the domain of Natural Language Processing (NLP) can be especially challenging due to the discrete nature of the input texts and the requirement to preserve both semantic coherence and grammatical correctness with the original texts (Zhang et al., 2020; Qiu et al., 2022; Goyal et al., 2023).

Generally, research studies of adversarial attacks can be classified as white-box, gray-box, or black-box. In a white-box approach, attackers can fully access the model architecture, weights, parameters,

or training datasets (Ebrahimi et al., 2018), whereas in the gray-box approach, the attackers have limited access to the model architecture (Xu et al., 2021). Finally, in the black-box approach, the attackers cannot access the model architecture but only query the model and get a prediction in return (Liu et al., 2017; Zhang et al., 2021).

A few common techniques for producing adversarial text examples have been addressed widely in the NLP field, such as character-level attacks (like inserting, removing, or swapping one or more characters within a word), word-level attacks (like inserting, removing, or replacing a word), and sentence-level attacks (like inserting, removing, or replacing a word or more than a word in a sentence) (Li et al., 2017; Ebrahimi et al., 2018; Gao et al., 2018; Liang et al., 2018; Feng et al., 2018). Some of these techniques can result in unnatural adversarial examples, making them easily distinguishable by humans, but recent research indicates that using rule-based synonym replacement strategies could generate adversarial text examples that appear more natural and similar to the original examples (Alzantot et al., 2018; Ren et al., 2019; Jin et al., 2020; Garg and Ramakrishnan, 2020; Li et al., 2020).

Only *two* papers addressed adversarial text attacks in Arabic, and both are character-level. Alshemali and Kalita (2019) proposed character-level adversarial attacks that rely on changing the morphological form of adjectives by adding one or more characters, which violates the noun-adjective agreement. An adjective is a word that describes a noun, and it must agree with the noun in definiteness (i.e., definite or indefinite), number (singular or dual), and gender (i.e., feminine or masculine). Alshemali and Kalita (2021) also proposed character-level attacks, which relied on the flip of one or two Arabic characters chosen based on non-native Arabic learners' most common spelling mistakes (usually incorrect use of visually similar characters). Both of these studies considered only the character-level adversarial examples and did not investigate the impact of transferability of these attacks among targeted models nor the effectiveness of defense mechanisms like adversarial training.

In this paper, we introduce the *first* study of word-level adversarial attacks in Arabic. We develop synonym-based word-level attacks using a Masked Language Modeling (MLM) task with an Arabic BERT model in a black-box manner against three state-of-the-art sentiment analysis

classifiers/models: BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019), WordCNN (word-based Convolutional Neural Networks) (Kim, 2014), and WordLSTM (Word-based Long Short-term Memory) (Hochreiter and Schmidhuber, 1997). We train these models on two available and large Arabic datasets, HARD (Hotel Arabic Reviews Dataset) (Elnagar et al., 2018) and MSDA (Sentiment Analysis for Social Media Posts in Arabic Dialect) (Boujou et al., 2021), to automatically generate adversarial text examples, attack these models using those generated adversarial text examples, and finally, assess the robustness of these models against adversarial text examples.

We also use human evaluation to evaluate the newly produced adversarial text examples using two criteria: grammatical similarity and semantic similarity. Furthermore, we study the transferability of these adversarial text examples generated by various models on different models studied and deeply investigate the effectiveness of the adversarial training defense mechanism on the BERT models against these adversarial text examples. Figure 1 illustrates the process of generating synonym-based word-level adversarial examples generation which we discuss in detail in Section 3.

Section 2 reviews related research works, while Section 3 delves into the methodology in more detail. Sections 4 and 5 discuss the automatic and human evaluations and their results. In Sections 6 and 7, we describe the transferability and defense mechanisms, along with their results. Lastly, in Sections 8 and 9, we conclude our paper by addressing its limitations and summarizing our contributions.

## 2 Related Work

Adversarial text attacks have been widely studied in the NLP field from different perspectives, like the attack setting (white-box, gray-box, and black-box), targeted space (embedding or input), and attack method (character-level, word-level, sentence-level) (Papernot et al., 2016; Li et al., 2017; Ebrahimi et al., 2018; Gao et al., 2018; Liang et al., 2018; Feng et al., 2018; Alzantot et al., 2018; Li et al., 2018; Ren et al., 2019; Li et al., 2020; Ribeiro et al., 2020; Zang et al., 2020; Jin et al., 2020; Li et al., 2021). Here, we will focus on related work that shares the same methodology as ours, taking advantage of the pre-trained models like the BERT model and its MLM training objective, but notably, they are all done only in English.

Jin et al. (2020) integrated two synonym replacement strategies (Alzantot et al., 2018; Ren et al., 2019) and proposed a baseline method called TEXTFOOLER to efficiently generate adversarial examples using synonym replacement techniques through word embeddings, ensuring the preservation of similar semantic meaning compared to the original words. Their findings show that pre-trained BERT models and other Deep Neural Networks (DNN) models were vulnerable to these adversarial text attacks, which could lead to misclassification or incorrect textual entailment predictions. The authors also emphasized improving the robustness of NLP models by incorporating defense mechanisms and testing their performance against adversarial attacks.

Several studies have proposed novel synonym replacement techniques using the MLM task with Large Language Models (LLMs) like BERT. For instance, Garg and Ramakrishnan (2020) proposed a novel method called BERT-based Adversarial Examples (BAE) for generating adversarial examples for text classification using MLM with BERT models. By utilizing pre-trained BERT’s ability to capture semantic meaning and context in text, BAE could generate adversarial examples more effectively than previous methods, avoiding detection by state-of-the-art classifiers. The authors also evaluated the effectiveness of BAE against various text classification models and demonstrated that it could generate robust adversarial examples. Li et al. (2020) proposed a practical method called BERT-Attack using MLM with BERT models to predict sub-word expansion, ensuring the generation of fluent and semantically preserved adversarial text examples. These adversarial examples successfully fooled the state-of-the-art models, such as the fine-tuned BERT models for various downstream tasks in NLP in a black-box manner. The authors evaluated the BERT-Attack’s effectiveness against BERT models in various text classification tasks and demonstrated its ability to reduce the accuracy of these models significantly. Lastly, Li et al. (2021) proposed an attack method called CLARE, a contextualized adversarial example generation model that could generate fluent and grammatically accurate outputs through a fill-in-mask procedure using MLM with RoBERTa models. CLEAR introduced three contextualized perturbations: replace, insert, and merge, which permit generating outputs of varying lengths, where it could flexibly integrate

these perturbations and apply them at any position in the inputs and then use them to attack the BERT models. The authors finally evaluated CLARE’s effectiveness against BERT models and demonstrated that CLARE achieved the best performance with the least modifications by combining all these three perturbations.

### 3 Methodology

#### 3.1 Datasets Used

In this work, we select two large Arabic datasets designed for text classification tasks: hotel reviews and sentiment analysis. We purposely chose one dataset written mostly in Modern Standard Arabic (MSA) and another written in Dialectical Arabic (DA) to observe how the models would behave when trained on different Arabic dialects.

- **Hotel Arabic Reviews Dataset (HARD)** is a balanced dataset with 93K hotel reviews written mostly in MSA, collected from Booking.com, and has four classes (Elnagar et al., 2018), which we remap to Poor, Fair, Good, and Excellent, instead of the original numerical labels (1, 2, 4, and 5).<sup>1</sup>

- **Sentiment Analysis for Social Media Posts in Arabic Dialect (MSDA)** is a balanced dataset that includes 50K posts written mostly in DA, collected from the X platform (formerly Twitter), and has three classes (Boujou et al., 2021): Positive, Neutral, and Negative.

Dataset	Avg Length	STD	Max Length
HARD	19.50	19.77	503
MSDA	9.99	9.46	326

Table 1: The general statistics of the two used datasets in terms of the Average Length (#words), Standard Deviation (STD), and Maximum Length (#words).

#### 3.2 Models Targeted

We train three deep learning classifiers/models that are widely used for text classification tasks: WordLSTM (Word-based Long Short-term Memory) (Hochreiter and Schmidhuber, 1997), Word-CNN (Word-based Convolutional Neural Network) (Kim, 2014), and BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019), on the HARD and MSDA datasets.

<sup>1</sup>Originally, labels 1 and 2 were negative, and 4 and 5 were positive. Notably, users were not given the choice of 3, only 5, 4, 2, or 1. We found this labeling confusing and remapped simply 1 to Poor, 2 to Fair, 4 to Good, and 5 to Excellent.

We use the same hyperparameters as Jin et al. (2020) used for the models in our study. We train WordCNN and WordLSTM models from scratch. For WordCNN models (Kim, 2014), we use three window sizes of 3, 4, and 5, and 100 filters for each window size. For the WordLSTM models, we use one bidirectional LSTM layer with 150 hidden states (Hochreiter and Schmidhuber, 1997). We train a GloVe (Global Vectors for Word Representation) model on each dataset to generate word vectors of size 200 dimensions for both models (Pennington et al., 2014) and use these GloVe embedding words to train the embedding layers in the WordCNN, and WordLSTM models on each dataset. For the BERT model, we begin with a pre-trained Arabic BERT model called AraBERT<sub>BASE</sub> v2<sup>2</sup> (Antoun et al., 2020), which has 12 layers with 768 hidden states, 12 heads, resulting in 136M trainable parameters, and we then fine-tune for text classification tasks using each dataset. We use different levels of data preprocessing before training WordLSTM and WordCNN models and fine-tuning BERT models due to the differences in their architectures, like the utilization of contextual embeddings in BERT models.

Table 2 shows the original accuracy (evaluation accuracy) of each model on each dataset. BERT models score the highest evaluation accuracies: 83% and 86% on HARD and MSDA, respectively. The WordLSTM model is not far behind, but the WordCNN models perform substantially worse.

Dataset	WordCNN	WordLSTM	BERT
HARD	75%	80%	83%
MSDA	77%	83%	86%

Table 2: Original accuracy (evaluation accuracy) of each model on each dataset (a test set of 10% of each dataset).

### 3.3 Adversarial Text Generation

The adversarial text generation task involves working with a dataset  $D$  in the form of  $(X, Y)$ , composed of pairs examples  $X$  and labels  $Y$  in the form of  $\{(x_1, y_1), \dots, (x_n, y_n)\}$ , alongside a black-box classifier  $C: X \rightarrow Y$ . We assume a soft label (with probability score) in a black-box setting, where the attacker can only query the classifier  $C$  for output labels  $Y$  and probabilities  $P$  given specific inputs, without access to any of the model’s parameters, weights, gradients, architecture, or training data.

<sup>2</sup>AraBERT<sub>BASE</sub> v2 model can be accessed here: <https://huggingface.co/aubmindlab/bert-base-arabertv2>.

Given an input example  $x$ , composed of  $W$  words in the form of  $(x = [w_1, w_2, w_3, \dots, w_n], y)$ , our goal is to create adversarial examples  $X_{ADV}$ , in such a way that  $C(X_{ADV}) \neq Y$ , meaning the prediction labels  $Y$  of  $C(X)$  do not equal the adversarial labels  $Y_{ADV}$  returned by  $C(X_{ADV})$ . Moreover, we aim for the adversarial examples  $X_{ADV}$  to exhibit grammatical correctness and maintain semantic similarity to the original inputs  $X$ .

#### 3.3.1 Adversarial Text Generation Steps

Here, we present the steps we use for generating adversarial examples  $X_{ADV}$ . We randomly select 1000 samples (original examples  $X$ ) from each dataset and process them following these steps:

**1) Word Importance Ranking:** We use the same scoring function ( $I_{w_i}$ ) as Jin et al. (2020) to measure the influence of a word  $w_i$ . Specifically, we quantify the importance of each token/word  $w_i$  in a sentence by deleting the tokens (one token each at a time) and calculating the prediction scores’ change before and after deleting that word  $w_i$ . We also clean the input original example  $x$  by removing the noise, emojis, stopwords, and punctuation marks using the NLTK Python library<sup>3</sup> before we feed the example  $x$  to the scoring function ( $I_{w_i}$ ) to reduce the computational overhead and ensure only words are fed to the scoring function ( $I_{w_i}$ ).

**2) Word Replacement Strategy:** We repetitively replace the most important words (one important word  $I_{w_i}$  at a time) in the input original example  $x$  using the MLM task with the BERT model to find synonym words for that important word  $I_{w_i}$ .

We use a different pre-trained version of AraBERT<sub>BASE</sub> v02<sup>4</sup> as our MLM model (Antoun et al., 2020) to generate the synonym words  $W_s$  and assigned the top  $K$  value to 50. After that, we use the CAMELBERT-CA POS-EGY model<sup>5</sup> (Inoue et al., 2021) as our Part-of-Speech (POS) tagger to ensure that the generated synonym words are grammatically correct. Next, we use a Sentence-Transformers model<sup>6</sup> to ensure the newly generated examples with the replaced synonym

<sup>3</sup>Natural Language Toolkit (NLTK): <https://www.nltk.org>.

<sup>4</sup>AraBERT<sub>BASE</sub> v02 model can be accessed here: <https://huggingface.co/aubmindlab/bert-base-arabertv02>.

<sup>5</sup>CAMELBERT-CA POS-EGY model can be accessed here: <https://huggingface.co/CAMEL-Lab/bert-base-arabic-camelbert-ca-pos-egy>.

<sup>6</sup>Sentence-Transformers model can be accessed here: <https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2>.



words are semantically similar to the original examples. We specifically used a multilingual pre-trained MPNet (Masked and Permuted Pre-training for Language Understanding) model (Song et al., 2020) trained on parallel data for 50+ languages, including Arabic (Reimers and Gurevych, 2020). We calculate the similarity score using the cosine similarity metric and set the similarity threshold to 0.80, as set by Jin et al. (2020). After the newly generated example passes the POS and similarity checkers, we finally have an attack candidate example derived from the original example  $x$ .

**3) Synonym BERT-based Attack:** Finally, we call an attack candidate example an ‘adversarial example’ ( $x_{adv}$ ) if it flips the prediction label  $y$  (the prediction label of the original example  $y$  before the attack  $\neq$  the prediction label of the attack candidate example  $y_{adv}$  after the attack). In other words, after replacing a word  $w$  with its synonym word  $w_s$  in the original example  $x$  using our synonym attack, the adversarial example  $x_{adv}$  force the targeted model/classifier to change its classification.<sup>7</sup>

## 4 Automatic Evaluation

### 4.1 Evaluation Metrics

We evaluate our synonym BERT-based attack using four metrics: Attack Success Rate (Att.<sub>SR</sub>), Accuracy Before Attack (Acc.<sub>BA</sub>), Accuracy After Attack (Acc.<sub>AA</sub>), and Attack Decrease Rate (Att.<sub>DR</sub>).

**1) Attack Success Rate (Att.<sub>SR</sub>)** is a metric designed to measure the successfulness of our synonym attack on a specific model and dataset, and it is calculated by dividing the number of adversarial examples produced by a model from a dataset by the total number of the randomly selected samples (original examples) multiplied by 100 for percentage normalization.

**2) Accuracy Before Attack (Acc.<sub>BA</sub>)** is calculated by taking the mean of prediction scores of the total number of selected samples input to the targeted model in a black box setting, meaning we only use the prediction scores instead of the targeted model’s original accuracy (evaluation accuracy). We believe taking the original accuracy of the model here is not a black-box manner because attackers are not supposed to know anything about the targeted model, including its original accuracy.

**3) Accuracy After Attack (Acc.<sub>AA</sub>)** is calculated by taking the mean of prediction scores of the total number of selected samples input to the targeted model in a black box setting after applying our synonym attack.

**4) Attack Decrease Rate (Att.<sub>DR</sub>)** is a metric designed to measure the effectiveness of our synonym attack on a specific model and dataset, and it is calculated simply by taking the difference between the Accuracy Before Attack (Acc.<sub>BA</sub>) and Accuracy After Attack (Acc.<sub>AA</sub>).

### 4.2 Evaluation Results

We first choose 1000 randomly selected examples from each dataset: HARD and MSDA, following Jin et al. (2020) and Li et al. (2021). Next, we evaluate our proposed synonym attack using metrics defined in subsection 4.1 above. The results of the automatic evaluations of our attack on each targeted model (WordCNN, WordLSTM, and BERT) and each dataset (HARD and MSDA) are displayed in Table 3. We find that our attack has successfully decreased the accuracies of the targeted models on each dataset, measured by the Acc.<sub>BA</sub> and Acc.<sub>AA</sub> metrics. For the 1000 randomly chosen examples, the BERT models fine-tuned on the MSDA and HARD datasets scored 90.55% and 88.59% as accuracies before our attack, and their accuracies after our attack have dropped to 63.62% and 73.90% on both datasets, respectively. It is clear that our attack strategy of substituting a word with a synonym works well because it creates a new example (adversarial example) that the targeted model has not encountered or seen before, forcing the targeted model to misclassify and cause a drop in its accuracy after the attack. Yet, we believe that the number of the evaluated examples, the different levels of data preprocessing, and the prediction misclassification rate of the block-box models are possible interpretations of the noticeable difference between the original accuracy mentioned in Table 2 and the Acc.<sub>BA</sub> in Table 3. Regardless of these constraints, our results demonstrate that all three models are indeed susceptible to our synonym attacks.

Furthermore, Table 3 summarizes the successfulness and effectiveness of our synonym BERT-based attack, measured by the Att.<sub>SR</sub> and Att.<sub>DR</sub> metrics, respectively. On the models level, our findings confirm that the DNN models (WordCNN and WordLSTM) are less susceptible to our attack than the BERT models. For instance, the attack decrease

<sup>7</sup>Appendix A provides some concrete examples of Arabic adversarial text examples generated using our synonym attack, along with their labels and their English translations.

Metric	WordCNN		WordLSTM		BERT	
	HARD	MSDA	HARD	MSDA	HARD	MSDA
Attack Success Rate (Att. <sub>SR</sub> )	50.00%	30.00%	51.00%	25.00%	51.00%	26.00%
Accuracy Before Attack (Acc. <sub>BA</sub> )	32.09%	45.15%	34.82%	47.48%	88.59%	90.55%
Accuracy After Attack (Acc. <sub>AA</sub> )	32.05%	39.31%	33.90%	41.73%	73.90%	63.62%
Attack Decrease Rate (Att. <sub>DR</sub> )	00.04%	05.84%	00.92%	05.75%	14.69%	26.93%

Table 3: Results of the attack success rate, accuracy before and after our attack, and attack decrease rate on each model and each dataset (the accuracies reported above are only for 1000 randomly selected examples).

rates of the WordCNN and WordLSTM models are both nearly 6% on the MSDA dataset, whereas the Att.<sub>DR</sub> of the BERT model on the same dataset is approximately 27%.

On the other hand, on the datasets level, we observe that our synonym attack is more successful and less effective on the HARD dataset than the MSDA dataset, meaning our attack on the HARD dataset produced more adversarial examples than on the MSDA dataset, but at the same time, these newly produced adversarial examples exhibit less impact on the targeted models trained on the HARD dataset. In contrast, our synonym attack generates fewer adversarial examples from the MSDA dataset, but those that succeed prove more potent. Notably, it is easier to successfully craft adversarial examples from the HARD dataset than the MSDA dataset since the MSDA dataset is a Dialectal Arabic (DA) dataset, and HARD is mostly a Modern Standard Arabic (MSA) dataset. Dialects in Arabic have fewer syntactic, morphologic, and orthographic rules than official Modern Standard Arabic (Grosvald et al., 2019).

## 5 Human Evaluation

### 5.1 Setup of Human Evaluation

We invite *four* human evaluators (all native Arabic speakers) to evaluate the naturalness of the Arabic adversarial text examples generated by our synonym attack. We randomly select 150 adversarial text examples (50 examples for each model from the HARD dataset)<sup>8</sup> to be evaluated by our human evaluators in terms of two major criteria: grammatical similarity and semantic similarity. As an inner-level evaluation, we ensure that two of these native Arabic evaluators have college degrees in the Arabic language (linguists), while the other two do not (non-linguists), to study the inner difference

<sup>8</sup>We only select generated adversarial examples from the HARD dataset because it is mostly written in Modern Standard Arabic, which can be easily evaluated in terms of Arabic grammar. MSA has syntactic, morphologic, and orthographic rules, not like the Dialectal Arabic (Grosvald et al., 2019).

in assessing the naturalness of our adversarial text examples between linguists and non-linguists, and ask them to evaluate all the selected examples.

For grammatical similarity assessment, we first retrieve the corresponding original examples to the randomly selected adversarial examples and separately group them into two groups: original and adversarial. We then task all the human evaluators to rate both groups anonymously, meaning we do not tell them which group is which to guarantee that the original examples do not influence human evaluators’ judgment. Inspired by Gagnon-Marchand et al. (2019), we use a 5-point Likert scale, where *one* represents *strongly incorrect*, *two* represents *incorrect*, *three* represents *correct to some extent*, *four* represents *correct*, and *five* represents *strongly correct* (Likert, 1932). Following Jin et al. (2020), we calculate the average score of the Likert scale measurements for each group (original and adversarial) and lastly divide the average score of the adversarial examples by the average score of the original examples of each human evaluator (linguists and non-linguists) to measure the precise grammatical similarity ratio between the original and adversarial examples.

For semantic similarity assessment, we accompany the randomly selected adversarial examples with their corresponding original examples and ask the human evaluators to rate whether the adversarial examples convey the same semantic meaning as the original examples. We use the same 5-point Likert scale, with different rating labels, ranging from *one* representing *strongly dissimilar* to *five* representing *strongly similar*. We then calculate the percentage of the average score of the Likert scale numbers (average score/number of rating labels) for each evaluator (linguists and non-linguists).

### 5.2 Human Evaluation Results

We observe in the grammatical similarity assessment that the non-linguist evaluators rated our adversarial examples and their original examples slightly higher than the linguist evaluators (except

<b>Evaluation Criteria</b>	<b>Human Evaluator</b>	<b>WordCNN</b>	<b>WordLSTM</b>	<b>BERT</b>
Grammatical Similarity	Linguists	92.00%	94.00%	98.00%
	Non-linguists	99.00%	95.00%	98.00%
	Overall Average	95.50%	94.50%	98.00%
Semantic Similarity	Linguists	89.00%	87.00%	91.00%
	Non-linguists	87.00%	86.00%	86.00%
	Overall Average	88.00%	86.50%	88.50%

Table 4: Results of human evaluation of our generated adversarial text examples from each targeted model (WordCNN, WordLSTM, and BERT) on the HARD dataset; no examples used from the dialectical MSDA dataset.

for the BERT model), whereas we have exactly the opposite results in the semantic similarity assessment (the linguists rated the examples higher than the non-linguists), as shown in Table 4. We assume that the gap in the background knowledge of the two human evaluators’ groups led to such results, where the non-linguists lack knowledge of Arabic syntax, making them rate the examples higher than the linguists in the grammatical similarity assessment. On the other hand, the deep understanding of the language and its semantics makes the linguists rate the examples higher than the non-linguists in the semantic similarity assessment.

Overall, as shown in Table 4, the human evaluation results (across all 4 evaluators) demonstrate that the adversarial text examples generated by our synonym attack is acceptable to Arabic native speakers, meaning that our adversarial examples preserve a similar level of grammatical correctness and convey similar semantic meaning. For example, the overall average scores ranged from 94.50% to 98% for grammatical similarity, whereas the average scores ranged from 86.50% to 88.50% for semantic similarity.

## 6 Transferability

### 6.1 Setup of Transferability

The transferability of an adversarial attack refers to its ability to reduce the accuracy of the targeted models (victim models) to a certain extent when attacked by the newly generated adversarial examples from other models (source models), where all the victim and source models trained on the same dataset (Szegedy et al., 2014; Goodfellow et al., 2015). To closely examine the transferability, we examine 245 adversarial examples from each of the HARD and MSDA datasets along with their corresponding original examples. Initially, we predict the accuracy of both the original examples and their corresponding adversarial examples using the victim models. We then calculate the difference (delta  $\Delta$ ) between the prediction accuracy of the

original examples and the adversarial examples, ultimately determining the transferability score for each model. We have not seen any other researcher using this delta difference method for the transferability of adversarial text attacks, even in English.

### 6.2 Transferability Results

Overall, first, we see that BERT (as the victim) has higher transferability scores than WordCNN or WordLSTM models, as shown in Table 5. This indicates that BERT is more vulnerable to transferred attacks. This result is similar to what Jin et al. (2020) saw in English. Second, we see that models trained in Dialectal Arabic (DA) are more vulnerable to transferred attacks than models trained in Modern Standard Arabic (MSA). Again, fewer generated adversarial examples from the dialectical MSDA dataset satisfy the synonym rules, but those that succeed prove more potent in the attack transferability between models.

## 7 Defense Mechanism

### 7.1 Setup of Defense Mechanism

We utilize adversarial training as a defense mechanism against our synonym attack, similar to Jin et al. (2020), and follow the approach introduced by Shrivastava et al. (2017). This method is widely adopted in image classification to enhance models’ robustness. To assess whether the employment of adversarial training enhances the robustness of these models, we add the generated adversarial examples to the original datasets. Then, we retrain the models and evaluate the robustness of these newly adversarially trained models. We only study the effectiveness of the adversarial training on BERT models. We collect adversarial examples from both datasets that successfully fooled BERT models and incorporate them into the original datasets to train the models adversarially. We then refine-tune these BERT models using the augmented datasets with adversarial examples and subject these adversarially fine-tuned models to our attacks again.

		Transferability Scores								
Datasets	Models	WordCNN (source)			WordLSTM (source)			BERT (source)		
		X	X <sub>ADV</sub>	Δ	X	X <sub>ADV</sub>	Δ	X	X <sub>ADV</sub>	Δ
HARD	WordCNN (victim)	—	—	—	52.65	47.34	5.31	65.71	34.28	31.43
	WordLSTM (victim)	56.32	43.67	12.65	—	—	—	60.81	39.18	21.63
	BERT (victim)	75.51	24.48	<b>51.03</b>	74.28	25.71	<b>48.57</b>	—	—	—
MSDA	WordCNN (victim)	—	—	—	87.34	12.65	74.69	86.53	13.46	73.07
	WordLSTM (victim)	83.26	16.73	66.53	—	—	—	82.04	17.95	64.09
	BERT (victim)	89.38	10.61	<b>78.77</b>	88.16	11.83	<b>76.33</b>	—	—	—

Table 5: Transferability of adversarial examples between victim and source models. Here,  $X$  refers to original examples, and  $X_{ADV}$  refers to adversarial examples. Let rows be  $N$  and columns be  $M$ , then cells  $NM$  are the accuracies of adversarial examples generated from the source model  $M$  and evaluated on the victim model  $N$ . A higher delta  $\Delta$  score indicates higher transferability between models.

## 7.2 Defense Mechanism Results

We find that BERT models’ adversarial training accuracies increased on both datasets, compared to their accuracies after the attack with no defense mechanism in place, as seen in Table 6. In other words, BERT models regain at least 2% in accuracy after applying adversarial training as a defense mechanism. These results agree with Jin et al. (2020)’s adversarial training results and provide further evidence that adversarial training is a good starting point for enhancing models’ robustness.

Metric	HARD	MSDA
Acc. Before Attack	88.59%	90.55%
Acc. After Attack	73.9%	63.62%
Adversarial Training Acc.	76.51%	65.69%

Table 6: Adversarial training results on BERT models.

## 8 Limitations

Due to the lack of availability of strong foundation models in Arabic and our lack of computational resources, one limitation of our attack is that it fundamentally depends on the pre-trained Arabic and multilingual models like the AraBERT model for retrieving synonym words, the CAMELBERT model for grammatical similarity checking, and the Sentence-Transformers model for semantic similarity checking, where their performance creates a bottleneck for the effectiveness of our text adversarial attack and the quality of our produced adversarial examples. Another limitation of our work is that the successfulness of our attack in generating adversarial examples is higher than its impact on the targeted models because the evaluation misclassification rate of the targeted models is another bottleneck of our attack, meaning if a model’s original (evaluation) accuracy score is 86% (e.g., BERT model on the MSDA dataset), then the misclassification rate of the model is 14%, which

undoubtedly affects the effectiveness of our attack negatively, especially since our attack setting is black-box, where it is impossible to remove these misclassified examples without compromising the attack setting (if we remove these misclassified examples, the attack setting will be a white-box).

## 9 Conclusion

In this work, we introduce the *first* Arabic synonym BERT-based adversarial attack using an MLM task with pre-trained BERT models against the state-of-the-art text classification models. We utilize two large Arabic datasets, namely HARD and MSDA, in a black-box manner. We find that BERT and other DNN models are generally susceptible to these Arabic adversarial examples, especially BERT models. We ask human evaluators to evaluate our produced adversarial examples using our attack in terms of grammatical similarity and semantic similarity and find that our attack generates examples that preserve semantic similarity and maintain Arabic grammar. We also study the transferability of these Arabic adversarial text examples by various source models on different victim models and observe that fine-tuned BERT models exhibit higher transferability when attacked by the other DNN models’ generated adversarial examples. Lastly, we investigate the effectiveness of the adversarial training defense mechanism on BERT models and find that the BERT models successfully regain at least 2% of their accuracies after applying the adversarial training as a defense mechanism.

## Reproducibility

We share our code scripts and trained models on GitHub at [https://www.github.com/NorahAlshahrani/bert\\_synonym\\_attack](https://www.github.com/NorahAlshahrani/bert_synonym_attack).



Dataset	Labels	Arabic Example	Translated Example
HARD	Original Label: Excellent	استثنائي ، فندق مريح جداً و كل شي كان جميل.	Exceptional , very comfortable hotel and everything was beautiful.
	Attack Label: Good	ممتاز ، فندق مريح جداً و كل شي كان جميل.	Excellent , very comfortable hotel and everything was beautiful.
	Original Label: Good	جيد ، استقبال جميل ونظافة وسهولة الوصول له وقريب من المطار والمراكز التسوق.	Good , beautiful reception, luxury, cleanliness, easy access, and close to the airport and shopping centers.
	Attack Label: Excellent	ممتاز ، استقبال جميل ونظافة وسهولة الوصول له وقريب من المطار والمراكز التسوق.	Excellent , beautiful reception, luxury, cleanliness, easy access, and close to the airport and shopping centers.
MSDA	Original Label: Negative	الله يرحم نادين ويخليك عيلتك. بعرف شو صعبة.	May God <b>have mercy on</b> Nadine and keep your family. I know how difficult it is.
	Attack Label: Positive	الله يحمي نادين ويخليك عيلتك. بعرف شو صعبة.	May God <b>protect</b> Nadine and keep your family. I know how difficult it is.
	Original Label: Positive	الأيام الحلوه كثيرا يا صاحبي ربنا يدبها.	Many good days! My friend, may God <b>perpetuate them</b> .
	Attack Label: Neutral	الأيام الحلوه كثيرا يا صاحبي ربنا موجود.	Many good days! My friend, God <b>is always there</b> .

Appendix A: Examples of the original Arabic and adversarial Arabic that are generated from the HARD and MSDA datasets and produced by the BERT models, along with their prediction labels and their English translations.

## Acknowledgments

We would like to thank Clarkson University and the Office of Information Technology (OIT) for providing computational resources and support that contributed to these research results. We also thank the anonymous evaluators who participated in the human evaluation study.

## References

- Basemah Alshemali and Jugal Kalita. 2019. [Adversarial Examples in Arabic](#). In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 371–376.
- Basemah Alshemali and Jugal Kalita. 2021. [Character-level Adversarial Examples in Arabic](#). In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 9–14.
- Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. [Generating natural language adversarial examples](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.
- Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. [AraBERT: Transformer-based Model for Arabic Language Understanding](#). In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.
- Anurag Arnab, Ondrej Miksik, and Philip HS Torr. 2018. [On the Robustness of Semantic Segmentation Models to Adversarial Attacks](#). In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 888–897.
- ElMehdi Boujou, Hamza Chataoui, Abdellah El Mekki, Saad Benjelloun, Ikram Chairi, and Ismail Berrada. 2021. [An Open Access NLP Dataset for Arabic Dialects: Data Collection, Labeling, and Model Construction](#). *arXiv preprint arXiv:2102.11000*.
- Nicholas Carlini and David Wagner. 2018. [Audio Adversarial Examples: Targeted Attacks on Speech-to-Text](#). In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7.
- Igino Corona, Giorgio Giacinto, and Fabio Roli. 2013. [Adversarial Attacks Against Intrusion Detection Systems: Taxonomy, Solutions and Open Issues](#). *Information Sciences*, 239:201–225.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#). In *North American Chapter of the Association for Computational Linguistics*.
- Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. [HotFlip: White-box adversarial examples for text classification](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.
- Ashraf Elnagar, Yasmin S. Khalifa, and Anas Einea. 2018. [Hotel Arabic-Reviews Dataset Construction for Sentiment Analysis Applications](#). In *Intelligent Natural Language Processing: Trends and Applications*, pages 35–52, Cham. Springer International Publishing.
- Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. [Pathologies of Neural Models Make Interpretations Difficult](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.
- Jules Gagnon-Marchand, Hamed Sadeghi, Md. Akmal Haidar, and Mehdi Rezagholizadeh. 2019. [SALSA-TEXT: Self Attentive Latent Space Based Adversarial Text Generation](#), page 119–131. Springer International Publishing.

- Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. [Black-Box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers](#). In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.
- Siddhant Garg and Goutham Ramakrishnan. 2020. [BAE: BERT-based Adversarial Examples for Text Classification](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.
- Yuan Gong and Christian Poellabauer. 2017. [Crafting Adversarial Examples for Speech Paralinguistics Applications](#). *arXiv preprint arXiv:1711.03280*.
- Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. [Explaining and Harnessing Adversarial Examples](#). *arXiv preprint arXiv:1412.6572*.
- Shreya Goyal, Sumanth Doddapaneni, Mitesh M. Khapra, and Balaraman Ravindran. 2023. [A Survey of Adversarial Defenses and Robustness in NLP](#). *ACM Computing Surveys*, 55(14s).
- Michael Grosvald, Sarah Al-Alami, Ali Idrissi, R Stockwell, M O’Leary, Z XU, and ZL Zhou. 2019. [Word Reading in Arabic: Influences of Diacritics and Ambiguity](#). In *36th West Coast Conference on Formal Linguistics*, pages 176–181. Cascadilla Proceedings Project.
- Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. 2014. [Deep Speech: Scaling up End-To-End Speech Recognition](#). *arXiv preprint arXiv:1412.5567*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long Short-Term Memory](#). *Neural Computation*, 9(8):1735–1780.
- Ling Huang, Anthony D. Joseph, Blaine Nelson, Benjamin I.P. Rubinstein, and J. D. Tygar. 2011. [Adversarial Machine Learning](#). In *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence, AISec ’11*, page 43–58, New York, NY, USA. Association for Computing Machinery.
- Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. [The Interplay of Variant, Size, and Task Type in Arabic Pre-trained Language Models](#). In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.
- Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. [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(05):8018–8025.
- Corey Kereliuk, Bob L Sturm, and Jan Larsen. 2015. [Deep Learning and Music Adversaries](#). *IEEE Transactions on Multimedia*, 17(11):2059–2071.
- Yoon Kim. 2014. [Convolutional Neural Networks for Sentence Classification](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.
- Jernej Kos, Ian Fischer, and Dawn Song. 2018. [Adversarial Examples for Generative Models](#). In *2018 IEEE security and privacy workshops (spw)*, pages 36–42. IEEE.
- Atul Kumar, Sameep Mehta, and Deepak Vijaykeerthy. 2017. [An Introduction to Adversarial Machine Learning](#). In *Big Data Analytics*, pages 293–299, Cham. Springer International Publishing.
- Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017a. [Adversarial Examples in the Physical World](#). *arXiv preprint arXiv:1607.02533*.
- Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017b. [Adversarial Machine Learning at Scale](#). *arXiv preprint arXiv:1611.01236*.
- Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2016. [Adversarial Machine Learning at Scale](#). *ArXiv*, abs/1611.01236.
- Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. [Contextualized Perturbation for Textual Adversarial Attack](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5053–5069, Online. Association for Computational Linguistics.
- Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. [TextBugger: Generating Adversarial Text Against Real-world Applications](#). *arXiv preprint arXiv:1812.05271*.
- Jiwei Li, Will Monroe, and Dan Jurafsky. 2017. [Understanding Neural Networks through Representation Erasure](#). *arXiv preprint arXiv:1612.08220*.
- Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. [BERT-ATTACK: Adversarial attack against BERT using BERT](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.
- Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. [Deep Text Classification Can Be Fooled](#). In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization.

- Rensis Likert. 1932. [A Technique for the Measurement of Attitudes](#). *Archives of psychology*.
- Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2017. [Delving into Transferable Adversarial Examples and Black-box Attacks](#). *arXiv preprint arXiv:1611.02770*.
- Flávio Mello. 2020. [A Survey on Machine Learning Adversarial Attacks](#). *Journal of Information Security and Cryptography (Enigma)*, 7(1):1–7.
- Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. [Practical Black-Box Attacks against Machine Learning](#). In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, ASIA CCS '17*, page 506–519, New York, NY, USA. Association for Computing Machinery.
- Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. 2016. [Crafting Adversarial Input Sequences for Recurrent Neural Networks](#). In *MILCOM 2016-2016 IEEE Military Communications Conference*, pages 49–54. IEEE.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global Vectors for Word Representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Nikolaos Pitropakis, Emmanouil Panaousis, Thanassis Giannetsos, Eleftherios Anastasiadis, and George Loukas. 2019. [A Taxonomy and Survey of Attacks Against Machine Learning](#). *Computer Science Review*, 34:100199.
- Shilin Qiu, Qihe Liu, Shijie Zhou, and Wen Huang. 2022. [Adversarial Attack and Defense Technologies in Natural Language Processing: A Survey](#). *Neurocomputing*, 492:278–307.
- Nils Reimers and Iryna Gurevych. 2020. [Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.
- Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. [Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.
- Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. [Beyond Accuracy: Behavioral Testing of NLP Models with CheckList](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.
- Saeid Samizade, Zheng-Hua Tan, Chao Shen, and Xiaohong Guan. 2020. [Adversarial Example Detection by Classification for Deep Speech Recognition](#). In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3102–3106. IEEE.
- Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. 2017. [Learning from Simulated and Unsupervised Images through Adversarial Training](#). In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2242–2251.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. [MPNet: Masked and Permuted Pre-training for Language Understanding](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.
- Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. [Intriguing Properties of Neural Networks](#). *arXiv preprint arXiv:1312.6199*.
- Rey Reza Wiyatno, Anqi Xu, Ousmane Dia, and Archy de Berker. 2019. [Adversarial Examples in Modern Machine Learning: A Review](#). *arXiv preprint arXiv:1911.05268*.
- Ying Xu, Xu Zhong, Antonio Jimeno Yepes, and Jey Han Lau. 2021. [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*, pages 4078–4087, Online. Association for Computational Linguistics.
- Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. [Word-level Textual Adversarial Attacking as Combinatorial Optimization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Piotr Żelasko, Sonal Joshi, Yiwen Shao, Jesus Vilalba, Jan Trmal, Najim Dehak, and Sanjeev Khudanpur. 2021. [Adversarial Attacks and Defenses for Speech Recognition Systems](#). *arXiv preprint arXiv:2103.17122*.
- Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. [Adversarial Attacks on Deep-learning Models in Natural Language Processing: A Survey](#). *ACM Transactions on Intelligent Systems and Technology*, 11(3).
- Yu Zhang, Kun Shao, Junan Yang, and Hui Liu. 2021. [Black-Box Universal Adversarial Attack on Text Classifiers](#). In *2021 2nd Asia Conference on Computers and Communications (ACCC)*, pages 1–5.