Adversarial Examples Against a BERT ABSA Model – Fooling Bert With L33T, Misspellign, and Punctuation,

Nora Hofer
nora.hofer@deepopinion.ai
DeepOpinion
Innsbruck, Austria

Pascal Schöttle
pascal.schoettle@mci.edu
Management Center Innsbruck
Innsbruck, Austria

Alexander Riezler
Sebastian Stabinger
firstname.lastname@deepopinion.ai
DeepOpinion
Innsbruck, Austria

ABSTRACT

The BERT model is de facto state-of-the-art for aspect-based sentiment analysis (ABSA), an important task in natural language processing. Similar to every other model based on deep learning, BERT is vulnerable to so-called *adversarial examples*, strategically modified inputs that cause a change in the model's prediction of the underlying input. In this paper, we propose three new methods to create character-level adversarial examples against BERT and evaluate their effectiveness on the ABSA task. Specifically, our attack methods mimic human behavior and use leetspeak, common misspellings, or wrongly-placed commas. By concentrating these changes on important words, we are able to maximize misclassification rates with minimal changes. To the best of our knowledge, we are the first to look into adversarial examples for the ABSA task and the first to propose these attacks.

CCS CONCEPTS

• **Security and privacy → Domain-specific security and privacy architectures**; • **Computing methodologies → Natural language processing**.

KEYWORDS

Natural Language Processing, BERT, ABSA, Adversarial Examples, Security

ACM Reference Format:
Nora Hofer, Pascal Schöttle, Alexander Riezler, and Sebastian Stabinger. 2021. Adversarial Examples Against a BERT ABSA Model – Fooling Bert With L33T, Misspellign, and Punctuation,. In *The 16th International Conference on Availability, Reliability and Security (ARES 2021), August 17–20, 2021, Vienna, Austria.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3465481.3465770

1 INTRODUCTION

Since their introduction in 2017, transformer-based language models took natural language processing (NLP) by storm and are widely used in numerous applications [27], including question answering and text classification. One of the most used transformer models is BERT (Bidirectional Encoder Representations from Transformers), which obtains state-of-the-art results in various tasks [4].

Sentiment Analysis (SA) is a common tool for identifying customer sentiment polarity towards a product or service by evaluating reviews [30]. However, sentiment alone provides only high-level insights and is not suitable for evaluating reviews on different products or services with more than one attribute. Aspect-based sentiment analysis (ABSA) is a fine-grained SA task that extracts both the aspects mentioned in a sentence and the sentiment associated with these aspects [20]. Despite its popularity, BERT's robustness against strategically manipulated inputs, occurring in realistic scenarios, is largely unexplored. This is highly concerning, given its increasing use in security-sensitive applications, such as fake news and hate speech detection [13, 19], e.g. in social media platforms and forums. In times of the COVID-19 pandemic, we once again see the importance of preventing the spread of misinformation on the Internet. This can be done, e.g., through a classification algorithm, as implemented by Twitter, that learns to detect untrustworthy information and labels the corresponding tweets to warn readers that they might be misinformed.

But, it has been shown that these detection algorithms can be circumvented by simple tricks. For example, an adversary can replace the letter O in the word COVID-19 with the number 0 and thus escape detection. In doing so, the recipients of the fake news would probably hardly notice afterwards that the sentence "*C0VID-19 can be cured by gargling salt water*" was deliberately manipulated.

In 2013, the term adversarial example was introduced to describe these scenarios where an adversary crafts inputs with the intention to cause a deep learning classifier to misclassify inputs [25]. Especially in the computer vision and image classification domain, powerful methods using gradient descent give the adversary the ability to mathematically optimize the attack and craft adversarial examples in the white-box setting, i.e. where the model's architecture and parameters are accessible to the adversary [3].

However, there are three main differences in attacking DNN models for computer vision and natural language processing:

(1) In an image, each pixel has a numerical representation within a fixed range and usually, pixels with similar numerical representations are closely related in terms of their characteristics. As textual data is symbolic, it is not possible to apply the same logic to the text domain. Here, increasing or decreasing the numerical representation of a word or sentence by the value of one might alter the complete meaning of a sentence.
(2) Adversarial images, usually bounded by an l_p-norm, supposedly preserve the image's semantic meaning. In the text domain changing a sentence's semantic happens easily. The

ARES 2021, August 17–20, 2021, Vienna, Austria

ACM ISBN 978-1-4503-9051-4/21/08…$15.00
https://doi.org/10.1145/3465481.3465770

sentence "I like cats" could be changed to "I like dogs" by replacing a single *word* or to "I like cars" by replacing a single *character*, both times obviously changing its semantic.

(3) Small modifications of image pixels are hardly recognized by human beings. Hence, if successful, adversarial examples will change the DNNs prediction but not human judgment. Small changes in the text, however, are easily detected and, therefore, render the possibility of attack failure. Moreover, a modification might also be corrected by spelling– or grammar–check systems.

It follows from these differences that, using gradient-based adversarial methods as in computer vision, for attacks in the text domain, can result in altered semantics, syntactically-incorrect sentences, or invalid words that cannot be matched with any words in the word embedding space [5].

Data augmentation is a method of generating additional, synthetic training data without acquiring, and labeling new data. In computer vision, this is commonly done by rotating, cutting, or flipping the input samples [28]. In the text domain, again, the process is a bit more complicated. A commonly known method is back-translation [21], where the training data is translated into any language, and translated back into the original language. This process often alters the sentence structure and/or replaces words with synonyms. Another established technique is called Easy Data Augmentation (EDA), consisting of synonym replacement, random insertion, random swap, and random deletion [29]. Random modification on the letter, word, or sentence level often generates data with incorrect grammar and invalid semantic meaning.

Previous efforts have shown that transformer models are vulnerable to adversarial examples in the white-box setting [8]. However, attacks in the black-box scenario (i.e. where an adversary can only access a model's output) seem more realistic.

Different variations of word-, or character level based perturbations have been proposed in the literature. Examples include replacing, deleting, swapping, or inserting words or characters [16], [5]. The disadvantage of word level perturbations is that an unintentional change of the semantic meaning of a sentence is likely to happen. This problem occurs less often when perturbing on the character level since the perturbed words are likely to remain the same.

In this work, we propose three different character-level adversarial attacks in the black-box setting against a BERT model fine-tuned on the ABSA task. Our contributions are:

(1) We propose three adversarial attack methods in the NLP domain, designed to be inconspicuous to human observers.
(2) We evaluate our proposed attack methods against the state-of-the-art language model BERT.
(3) To the best of our knowledge, we are the first to research adversarial examples against a deep learning model fine-tuned on the ABSA task in the black box setting.

The remainder of the paper is organized as follows: Section 2 recaps related work before we describe our attack methods in Section 3. The experimental setup is described in Section 4 and the results in Section 5. Section 6 concludes the paper.

2 RELATED WORK

Due to the discrete nature of the text domain and the prerequisite to retain semantic and grammar, gradient-based attack methods, as commonly used in computer vision [9, 18, 25], cannot be easily adapted here. Nonetheless, several ways of creating adversarial examples in the text domain have been proposed, including replacing, deleting, swapping, or inserting on a character, word, or sentence level [11, 12, 24]. Previous works have conducted attacks in the black- and white-box scenario [5, 7] addressing numerous different text classification and generation tasks. Furthermore, [14] proposed adversarial training for a model addressing the ABSA task in the white-box setting.

As transformer-based language models have achieved state-of-the-art results in various NLP tasks, the robustness of these models was challenged by various researchers lately. Their results show that those models are vulnerable to attacks in a white box setting [24]. These approaches, however, may lack practical relevance since they often result in incorrect grammar and altered semantic, and therefore require human revision. Other approaches [12, 17, 31] research the black-box setting, however, using word-level attacks that still require human revision as unintentional changes of the semantic meaning of a sentence are likely to happen here. Adversarial examples on the character-level mimic a more realistic scenario and prevent the alteration of grammar and semantic meaning.

Instead of directly using the model's gradients' signs or values, in the text domain, different search methods have been introduced to facilitate the crafting of adversarial examples. Those include greedy- and beam search [5, 7] for word importance ranking, as well as genetic algorithms [1]. Another approach is the leave-one-out method (LOO). Here, iteratively, each word of an input sequence is removed once, and the predicted output label of the remaining sentence is compared with the original sentence's output label. This simple approach for identifying target words for adversarial modifications was shown to be successful [17].

3 PROPOSED ATTACKS

We draw on existing work and use the LOO method to determine zero to many important words for each input sequence to execute three different attacks that craft examples mimicking user-generated content.

The effort to create a successful adversarial example can be minimized by targeting the attack on the word with the strongest influence on the prediction outcome.

3.1 Design Criteria

Our attack methods overcome the described limitations of creating adversarial examples in the text domain. By design, we have opted for perturbation methods on the character level that most likely do not alter an input sequence's semantic meaning or grammar. Even under circumstances where the change of a single character results in a new, existing word of the dictionary, with a different semantic meaning, we consider the example valid, since this scenario could also happen in a real world, e.g., as a typo. All our adversarial changes are supposed to prevent humans from easily spotting them. As we believe that practical relevance is important for research in adversarial machine learning, we pursued the following objectives:

(1) keeping semantic meaning of the input data
(2) inconspicuousness to a human observer
(3) relevance in a real-world scenario.

These are the main points in which our work differs from previous ones.

3.2 Attack Methods

We describe our three proposed attack methods and highlight them exemplarily, given the laptop review *"It's wonderful for computer gaming"*. Note that the determined important word is "*wonderful*". In the following examples, induced character changes are visualized by underlining the respective characters.

Leetspeak (1337) is characterized by the use of non-alphabet characters to substitute one or multiple letters of one word with visually similar-looking symbols, so-called homoglyphs. Commonly used homoglyphs in leetspeak are numbers[1]. We generate adversarial examples by swapping the letters **a, e, l, o,** and **s** of the identified important words with the numbers **4, 3, 1, 0,** and **5**, respectively. The visual similarity of the chosen letters and numbers keeps the word legible to humans and preserves the semantic meaning of the word. Note that a modified important word can theoretically contain as many numbers as it has letters. The leetspeak attack applied on the example review results in the modified input sequence *"It's w0nd3rfu1 for computer gaming"*. In online domains, humans do not find the usage of leetspeak suspicious. Thus, adversarial examples generated with this method will appear legitimate in those situations.

Misspelling Inspired by [24], we use a list of common misspellings from Wikipedia[2] to generate adversarial examples. We first determine the important words and then replace them with all possible misspellings. The list consists of 4 282 entries, where one word can have multiple misspelling variations. The resulting modified example sentence is *"It's wonderfull for computer gaming"*. Also here, the semantic meaning of the modified word is preserved and the modification is unobtrusive to a reader.

Punctuation The results from [6] suggest that BERT is robust to changes in irrelevant punctuation marks. We believe their results call for further research and want to find out whether a single comma added after the important word poses an efficient way to cause misclassifications when addressing the ABSA task using BERT. One additional comma is unobtrusive, might occur in practical use cases, and is not easily identified as an adversarial example by a human observer. Perturbing the example sentence using the punctuation method results in *"It's wonderful, for computer gaming"*.

4 EXPERIMENTAL SETUP

4.1 Data Set

We conduct our experiments on the laptop dataset of the SemEval-2015 Task 12: Aspect Based Sentiment Analysis [22], which is considered a benchmark dataset for research on the ABSA task. An aspect category is defined as a combination of an entity (e.g., laptop) and an attribute describing the entity (e.g., price). The second part of the annotation is the sentiment label which expresses the polarity towards the aspect category and can take on the values *positive, neutral* or *negative*. The laptop dataset comes with a standard training and testing data split and consists of 1 739 sentences in the training data and 761 sentences in the test data.

4.2 Model

The target model for this work is the pre-trained transformer model BERT [4]. In a first step, we fine-tune a BERT base model[3] on the laptop domain and on the ABSA task in a second step. The BERT language model is trained in a self-supervised way on a large, domain specific, and unlabeled corpus solving the tasks masked language modeling (MLM) and next sentence prediction (NSP). We employ the Amazon Laptop reviews dataset [10] to have sufficient training data and use Adam for optimization [15]. We fine-tune the language model using a batch size of 32 and a learning rate of $3 \cdot 10^{-5}$ with random initialization. The input sequence length is 256 tokens, resulting in four sentences per sequence on average. Due to the relatively small number of training examples in the corpus, we fine-tune BERT base for 30 epochs, such that the model sees about 30 million sentences during training. That way, a single sentence appears multiple times. We use the code from [23], which can also be found on GitHub[4]. To fine-tune our model on the laptop domain, we use unlabeled Amazon Laptop reviews [10]. To avoid training bias for the SemEval-2015 test data, we filter out reviews that appeared in both the Amazon and the SemEval-2015 Test Dataset. Moreover, we remove reviews that contain less than two sentences from the training corpora to achieve compatibility with the next sentence prediction task used for fine-tuning. After the text pre-processing, there are 1 007 209 unlabeled sentences left in the corpus. For fine-tuning the pre-trained BERT model on the ABSA task, we use the Ranger optimizer [32] and set the learning rate to $3 \cdot 10^{-5}$. We use a batch size of 32 and fine-tune it for 20 epochs. For tokenization, we used the PreTrainedTokenizer introduced with the BERT base model[5]. Parameter choices for batch size, learning rate and number of epochs are based on the experiments mentioned above[6]. The accuracy of the model is 79.8%.

4.3 Evaluation metrics

Following the literature, we consider an adversarial example successful if we are able to change the prediction of an input sentence. In the ABSA task, we consider a prediction as changed if either

a) a different entity, attribute, or sentiment is predicted,
b) the model does no longer predict any aspect, or
c) the model predicts an aspect where it did not predict one before.

We measure the effectiveness of our three attack methods using a distinct and an overall attack success rate and summarize the results in Table 1.

First, we filter the SemEval 2015 dataset for unique items (Dataset A), which results in 943 sentences. Then, we use the LOO method (see Sec. 3) to detect important words for all the sentences in Dataset A. Note that a sentence can have more than one important word.

[1] https://en.wikipedia.org/wiki/Leet
[2] https://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings
[3] https://huggingface.co/bert-base-uncased
[4] https://github.com/deepopinion/domain-adapted-atsc
[5] https://huggingface.co/transformers/main_classes/tokenizer.html
[6] https://github.com/deepopinion/domain-adapted-atsc

Perturbation Method	Leetspeak	Misspellings	Punctuation
Dataset A - # of original sentences	943	943	943
Dataset B - # of modifiable original sentences	897	369	943
Dataset C - # of adversarial sentences	2232	1354	2555
Dataset D - # of changed predictions total	1066	420	382
Dataset E - # of changed predictions per sentence	790	259	253
Overall Success Rate	**47.76%**	**31.01%**	**14.95%**
Distinct Success Rate	**88.07%**	**70.19%**	**26.83%**

Table 1: Comparison of the success rates of the three attack methods.

Among these, we identify for each of the three proposed attacks modifiable words. For one important word, there may be none, or one modifiable word per attack. Sentences containing modifiable words are collected in Dataset B and differ per attack method. For the Punctuation attack, which adds a comma behind the targeted word, all important words are modifiable, which results in the same number of sentences in Dataset A and B.

In the next step, we create all possible adversarial examples from Dataset B, e.g., all possible misspellings of all identified important words result in different elements in the dataset of adversarial sentences (Dataset C).

Finally, we use the BERT model to predict aspect and sentiment of all elements from Dataset C and compare the prediction to the original one. If the prediction changes (in any of the ways described above), the respective sentence is added to the adversarial datasets D and E. While Dataset D contains all sentences from Dataset C, where the prediction was changed, Dataset E contains all sentences from Dataset B (one entry per sentence). The overall attack success rate is calculated as the ratio $|\text{Dataset D}|/|\text{Dataset C}|$. The distinct attack success rate is calculated as the ratio $|\text{Dataset E}|/|\text{Dataset B}|$.

5 RESULTS AND DISCUSSION

In this section, we present the results of our work. The full results and code are available on GitHub[7].

5.1 Results

Our final results are summarized in the bottom line of Table 1.

An attack success rate of 100% would mean that every sentence containing an important word modified by the method caused the algorithm to change its prediction. The most successful method is Leetspeak, with achieves an overall success rate of 47.8% and a distinct success rate of 88%. By using misspellings, we generated incorrect predictions for 31% of all modified sentences and 70% of the sentences in the initial dataset. Simply inserting one additional comma behind the important word caused the model to change its prediction for 15% of all modified sentences and 27% of the sentences in the initial dataset. In the case of our example sentence *"It's wonderful for computer gaming"*, we were able to change the result of the prediction by using any of the three methods. Table 2 shows more example sentences. Depending on whether none or several important words, or none or several modification variants are possible, sentences do not appear, or appear several times per attack.

The complete list of successful adversarial sentences can be found in our GitHub repository. A statement about common class confusions caused by adversarial sentences depends entirely on the input data as well as the defined aspects. Therefore, no general conclusion could be drawn from such a statement.

5.2 Discussion

Our experiments demonstrate that BERT can be fooled by input modifications on the character level, imitating real-world scenarios in the black-box setting. All three attack methods cause the classifier to change its predictions. DNN-based text classification continuously gains importance for enhancing the safety of users, e.g., in online forums or social media [2], where leetspeak is commonly used. The findings of our experiments using character substitution call for action to increase safety in those environments. Although the comparison of the attack success rates reveals that the Punctuation method was the least effective, 15% attack success is still not negligible, especially taking into account that the method only adds a single comma. Further inspecting these results, we found that the BERT tokenizer separates punctuation marks from the preceding words, thus our method produces a new token but does not change the token representation of the determined important word. Contrary to [6], we found that BERT is indeed sensitive to irrelevant punctuation marks if positioned behind the important word. This deviation in results calls for further research on the positioning of the punctuation marks. BERT is pre-trained on Wikipedia and a huge book corpus [33], containing over 10 000 books of different genres. Since the model has seen at least some of the misspellings (remember that our list of common misspellings also comes from Wikipedia) during the pre-training process, one would assume that common misspellings do not significantly affect the predictions. A success rate of 31% indicates that the opposite is true. This falls in line with [24], who show the vulnerability of BERT against different typo methods for sentiment analysis and question answering.

Finally, for now, we did not differentiate between adversarial examples that change only the aspect but keep the sentiment and those that change the sentiment or predict a sentiment where there was none predicted before. Depending on the application scenario, such a differentiation probably makes sense in future work.

[7] https://github.com/NoraH2004/adv_absa

	Sentence	ENTITY#ATTRIBUTE	Sentiment
Original	i would really recommend to any person out there to get this laptop cause its really worth it.	LAPTOP#GENERAL	POS
Leet Speak	i would really recommend to any person out there to get this laptop cause its really w0rth it.	LAPTOP#GENERAL *LAPTOP#QUALITY*	POS NEG
Misspellings	-	-	-
Punctuation	i would really recommend to any person out there to get this laptop cause its really worth, it.	LAPTOP#GENERAL *LAPTOP#PRICE*	POS NEG
Original	It's more expensive but well worth it in the long run.	LAPTOP#GENERAL LAPTOP#QUALITY LAPTOP#PRICE	POS NEG NEG
Leet Speak	It's more expensive but well worth it in th3 long run.	LAPTOP#GENERAL ~~LAPTOP#QUALITY~~ LAPTOP#PRICE	POS ~~NEG~~ NEG
Misspellings	It's more expensive but well worth it in teh long run.	LAPTOP#GENERAL ~~LAPTOP#QUALITY~~ LAPTOP#PRICE	POS ~~NEG~~ NEG
Punctuation	It's more expensive but well worth it in the, long run.	LAPTOP#GENERAL ~~LAPTOP#QUALITY~~ LAPTOP#PRICE	POS ~~NEG~~ NEG
Punctuation	It's more expensive but well, worth it in the long run.	LAPTOP#GENERAL ~~LAPTOP#QUALITY~~ LAPTOP#PRICE	POS ~~NEG~~ NEG
Original	After a little more than a year of owning my MacBook Pro, the monitor has completely died.	DISPLAY#OPERATION_PERFORMANCE DISPLAY#QUALITY	NEG NEG
Leet Speak	After a little more than a year of owning my MacBook Pro, the monitor has completely di3d.	~~DISPLAY#OPERATION_PERFORMANCE~~ DISPLAY#QUALITY *DISPLAY#DESIGN_FEATURES* *GRAPHICS#QUALITY* *GRAPHICS#DESIGN_FEATURES*	~~NEG~~ NEG *NEG* *NEG* *NEG*
Misspellings	After a little more than a year of owning my MacBook Pro, tje monitor has completely died.	~~DISPLAY#OPERATION_PERFORMANCE~~ DISPLAY#QUALITY	~~NEG~~ NEG
Punctuation	After a little more than a year of owning my MacBook Pro, the monitor, has completely died.	~~DISPLAY#OPERATION_PERFORMANCE~~ DISPLAY#QUALITY	~~NEG~~ NEG

Table 2: Example sentences modified using the three attack methods Leet Speak, Misspellings, and Punctuation. The modified important words are underlined. Labels that were no longer predicted through the modification are shown as crossed out. Newly added are indicated by cursive font.

Adversarial (re)training poses an important defense strategy against adversarial attacks, where a generated adversarial dataset is used to train the target model in order to increase its robustness [26]. An examination of potential differences, whether the adversarial dataset is used for pre-training or fine-tuning should be tested in further experiments. The pre-training with our generated data set could contribute to further findings in the field.

Compared to data augmentation, where semantics, grammar, and inconspicuousness are not a priority, we were able, through the careful design of the attack methods, to generate samples that are valid and do not change the human judgment, yet cause the classifier to produce false output labels. Obviously it is not possible to fully rule out that our attacks change human judgement. In our paper we refrained from an additional analysis, since a manual, generously sampled review of the data showed that such a review is not necessary, and would not change the message of the paper. In order to prevent the occurrence of cases where the human decision is changed, a quantitative study, e.g. in the form of a questionnaire, should be conducted.

6 CONCLUSION

In this work, we study adversarial attacks against the state-of-the-art BERT model addressing the ABSA task. When designing our attack methods, we have placed great emphasis on the practical relevance by generating perturbations on the character-level, that could have been produced exactly as they are by human beings. Thus, the adversarial examples we create are not perceived as such easily and do not change the semantic meaning or grammar. Furthermore, we conduct our attacks in the black-box setting, where we do not have access to the model's architecture or parameters. Our results demonstrate a general vulnerability and show that simple input modifications, likely to happen in a real-world scenario, are sufficient to fool a state-of-the-art NLP model.

Summarizing our results, we have shown that using leetspeak, we were able to change almost 50% of the model's predictions. Using common misspellings, we fooled the model in more than 30% of the cases, and by simply inserting a comma after the important word, 15% of predictions have changed.

One established result in adversarial machine learning for computer vision is that a wide variety of models with different architectures misclassify the same adversarial examples, even when trained

on different subsets of training data [9]. Testing our generated adversarial datasets on other language models as a next step would provide information about the "transferability" of our attacks. Additionally, established countermeasures, such as adversarial training [18] should be further investigated for their effectiveness in the text domain. Our result dataset can be used in the process in order to increase the robustness against such attacks. This paper is intended to raise awareness about the potential vulnerability of the BERT model and encourages to not entirely rely on these models for security relevant tasks, such as the detection of hate speech or false information. The different characteristics of our attacks provide additional information on the vulnerability of the transformer models.

Finally, we want to mention that we choose the title as it is intentionally, including the misspelling of "misspelling" and the comma at the very end, to highlight leetspeak, misspellings, and additional punctuation, the basis for our proposed attacks.

ACKNOWLEDGMENTS

The second author was supported by the Austrian Science Fund (FWF) under grant no. I 4057-N31 ("Game Over Eva(sion)").

REFERENCES

[1] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani B. Srivastava, and Kai-Wei Chang. 2018. Generating Natural Language Adversarial Examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics, 2890–2896. https://doi.org/10.18653/v1/d18-1316

[2] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep Learning for Hate Speech Detection in Tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*. 759–760.

[3] Nicholas Carlini and David A. Wagner. 2017. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, Bhavani M. Thuraisingham, Battista Biggio, David Mandell Freeman, Brad Miller, and Arunesh Sinha (Eds.). ACM, 3–14. https://doi.org/10.1145/3128572.3140444

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[5] 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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*. Association for Computational Linguistics, 31–36. https://doi.org/10.18653/v1/P18-2006

[6] Adam Ek, Jean-Philippe Bernardy, and Stergios Chatzikyriakidis. 2020. How does Punctuation Affect Neural Models in Natural Language Inference. In *Proceedings of the Probability and Meaning Conference (PaM 2020)*. 109–116.

[7] 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, SP Workshops 2018, San Francisco, CA, USA, May 24, 2018*. IEEE Computer Society, 50–56. https://doi.org/10.1109/SPW.2018.00016

[8] Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based Adversarial Examples for Text Classification. *CoRR* abs/2004.01970 (2020). arXiv:2004.01970

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

[10] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends With One-Class Collaborative Filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.

[11] Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328* (2017).

[12] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is BERT Really Robust? Natural Language Attack on Text Classification and Entailment. *arXiv preprint arXiv:1907.11932* 2 (2019).

[13] Heejung Jwa, Dongsuk Oh, Kinam Park, Jang Mook Kang, and Heuiseok Lim. 2019. exBAKE: Automatic Fake News Detection Model Based on Bidirectional Encoder Representations from Transformers (BERT). *Applied Sciences* 9, 19 (2019), 4062.

[14] Akbar Karimi, Leonardo Rossi, Andrea Prati, and Katharina Full. 2020. Adversarial Training for Aspect-Based Sentiment Analysis With BERT. *arXiv preprint arXiv:2001.11316* (2020).

[15] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

[16] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. TextBugger: Generating Adversarial Text Against Real-world Applications. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society. https://www.ndss-symposium.org/ndss-paper/textbugger-generating-adversarial-text-against-real-world-applications/

[17] 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 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 6193–6202. https://doi.org/10.18653/v1/2020.emnlp-main.500

[18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

[19] Alex Nikolov and Victor Radivchev. 2019. Nikolov-Radivchev at SemEval-2019 Task 6: Offensive Tweet Classification With Bert and Ensembles. In *Proceedings of the 13th International Workshop on Semantic Evaluation*. 691–695.

[20] Ioannis Pavlopoulos. 2014. Aspect Based Sentiment Analysis. *Athens University of Economics and Business* (2014).

[21] Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette De Buy Wenniger, and Peyman Passban. 2018. Investigating backtranslation in neural machine translation. In *EAMT 2018 - Proceedings of the 21st Annual Conference of the European Association for Machine Translation (EAMT 2018 - Proceedings of the 21st Annual Conference of the European Association for Machine Translation)*, Juan Antonio Perez-Ortiz, Felipe Sanchez-Martinez, Miquel Espla-Gomis, Maja Popovic, Celia Rico, Andre Martins, Joachim Van den Bogaert, and Mikel L. Forcada (Eds.). European Association for Machine Translation, 249–258. 21st Annual Conference of the European Association for Machine Translation, EAMT 2018 ; Conference date: 28-05-2018 Through 30-05-2018.

[22] Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 Task 12: Aspect Based Sentiment Analysis. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*. 486–495.

[23] Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. 2019. Adapt or Get Left Behind: Domain Adaptation Through Bert Language Model Finetuning for Aspect-Target Sentiment Classification. *arXiv preprint arXiv:1908.11860* (2019).

[24] Lichao Sun, Kazuma Hashimoto, Wenpeng Yin, Akari Asai, Jia Li, Philip Yu, and Caiming Xiong. 2020. Adv-BERT: BERT Is Not Robust on Misspellings! Generating Nature Adversarial Samples on BERT. *arXiv preprint arXiv:2003.04985* (2020).

[25] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing Properties of Neural Networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1312.6199

[26] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. 2018. Ensemble Adversarial Training: Attacks and Defenses. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rkZvSe-RZ

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in neural information processing systems*. 5998–6008.

[28] Jason Wang, Luis Perez, et al. 2017. The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit* 11 (2017).

[29] Jason Wei and Kai Zou. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 6382–6388. https://doi.org/10.18653/v1/D19-1670

[30] Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. 2003. Sentiment Analyzer: Extracting Sentiments About a Given Topic Using Natural Language Processing Techniques. In *Third IEEE international conference on data mining*. IEEE, 427–434.

[31] 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, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 6066–6080.

[32] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. 2019. Lookahead Optimizer: k Steps Forward, 1 Step Back. In *Advances in Neural Information Processing Systems*. 9597–9608.

[33] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 19–27. https://doi.org/10.1109/ICCV.2015.11