ADVERSARIAL EXAMPLES AGAINST A BERT ABSA MODEL – FOOLING BERT WITH L33T, MISSPELLIGN, AND PUNCTUATION,

N. Hofer *A. Rietzler* *S. Stabinger*

DeepOpinion
Research & Development
Innsbruck, Austria

P. Schöttle[*]

Management Center Innsbruck
Digital Business & Software Engineering
Innsbruck, Austria

ABSTRACT

The BERT model is de facto state-of-the-art for aspect-based sentiment analysis (ABSA), one of the most important tasks 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.

***Index Terms*—** Natural Language Processing, BERT, ABSA, Adversarial Examples, Security

1. INTRODUCTION

Since their introduction in 2017, transformer-based language models took natural language processing (NLP) by storm and are widely used in various applications [1]. One of the most used transformer models is BERT, which obtains state-of-the-art results in various tasks [2]. Despite its popularity, the model's security against strategically manipulated inputs existing in realistic scenarios is largely unknown. This is highly concerning, given its increasing use in security-sensitive applications, such as fake news and hate speech detection [3, 4]. In 2013, the term "adversarial example" was introduced to describes scenarios where an adversary crafts inputs with the intention to cause a deep neural network to change its classification output [5]. Previous efforts have shown that transformer models are vulnerable to adversarial examples in the white-box setting [6], a case where the model's architecture and parameters are accessible to the adversary. However, attacks in the black-box scenario, where an attacker usually can only access a model's output, seem more realistic.

Sentiment Analysis (SA) is a common tool for identifying customer sentiment polarity towards a product or service by evaluating reviews [7]. 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 the aspects [8].

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

1. 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.
2. We propose three new adversarial attack methods in the NLP domain, which, by design, are inconspicuous to a human observer.
3. We evaluate our proposed attack methods against the state-of-the-art language model BERT.

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 [5, 9, 10], cannot be easily adapted here. Nonetheless, several ways of creating adversarial examples in the text domain have been proposed, including replacing, deleting, swapping, or insertions on a character, word, or sentence level [11, 12, 13]. Previous works have conducted attacks in the black- and

[*]Parts of this research was funded by Austrian Science Fund (fwf) under grant no. I 4057-N31.

white-box scenario [14, 15] addressing numerous different text classification and generation tasks. Furthermore, [16] proposed adversarial training for a model addressing the ABSA task in the white-box setting. Since 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 [11]. 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, 18] research the black-box setting, however, use 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 [14].

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 [14, 15] for word importance ranking, as well as genetic algorithms [19]. Another important approach, which is used in this paper in the leave-one-out method (LOO), where in an iterative process, 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. Note that one input sequence can contain $[0, 1, \ldots, n]$ important words. This simple approach for identifying target words for adversarial modifications works has proven successful [18].

3. PROPOSED ATTACKS

We draw on existing work and use the LOO method to determine $0, \ldots, n$ important words for each input sequence to execute three different attacks that craft examples mimicking user-generated content.

3.1. Design Criteria

All adversarial changes are supposed to prevent humans from easily spotting them. By design, we have opted for perturbation methods on the character level that do not alter an input sequence's semantic meaning or grammar since the perturbed words remain the same. Even under circumstances where the change of a single character randomly 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 setting.

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*", 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. 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"*.

Misspelling Inspired by [11], 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"*.

Punctuation The results from [20] 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. Model (BERT)

The target model for this work is the pre-trained transformer model BERT [2]. 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 unlabeled corpus solving the tasks masked language modeling (MLM) and next sentence prediction (NSP). We employ the Amazon Laptop reviews dataset [21] to have sufficient training data and use Adam for optimization [22]. 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,

[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

resulting in four sentences per sequence on average. Due to the relatively low number of training data, 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. The code can be found on GitHub[4][23]. For fine-tuning the pre-trained BERT model on the ABSA task, we use the Ranger optimizer [24] 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].

4.2. Data Set

The experiments are conducted on the laptop dataset of the SemEval-2015 Task 12: Aspect Based Sentiment Analysis [25], which is considered the 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., durability). 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.

To fine-tune BERT on the laptop domain, we used unlabeled Amazon Laptop reviews [21]. To avoid training bias for the SemEval-2015 test data, we filtered out reviews that appeared in both the Amazon and the SemEval-2015 Test Dataset. Moreover, we removed reviews that contain less than two sentences from the training corpora to achieve compatibility with the NSP task used for fine-tuning. After the text pre-processing, there are 1,007,209 unlabeled sentences left in the corpus.

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 efficiency of our three attack methods using the attack success rate and compare 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. 2) to detect important words for all the sentences in Dataset A. Among these, we identify for each of the three proposed attacks modifiable words. Sentences containing modifiable words are collected in Dataset B and differ per attack method.

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 used the BERT model to predict aspect and sentiment of all elements from Dataset C and compared the prediction to the original one. If the prediction changed, the respective sentence was added to Dataset D. The overall attack success rate is calculated as the ratio $|\text{Dataset D}|/|\text{Dataset C}|$.

5. RESULTS AND DISCUSSION

In this section, we present selected results of our work. The full results and code are available on GitHub[6].

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 attack success rate of 47.8%. By using misspellings, we generated incorrect predictions for 31%. Simply inserting one additional comma behind the important word caused the model to change its prediction for 15% of the sentences. 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.

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 [26], 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 [20], we found that BERT is indeed sensitive to irrelevant punctuation marks if

[4]https://github.com/deepopinion/domain-adapted-atsc

[5]https://huggingface.co/transformers/main_classes/tokenizer.html

[6]https://github.com/NoraH2004/adv_ABSA

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 through modification	1066	420	382
Attack Success Rate	**0.4776**	**0.3101**	**0.1495**

Table 1. Comparison of the three attack methods

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 [27], containing over 10 000 books of different genres. Since the model has seen at least some of the misspellings 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. Our findings fall in line with [11], 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.

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 by humans easily and do not change the semantic meaning. Furthermore, we conduct our attacks in the black-box scenario, 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 [10]. 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 [9] should be further investigated for their effectiveness in the text domain.

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

7. REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.

[3] H. Jwa, D. Oh, K. Park, J. M. Kang, and H. Lim, "exbake: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert)," *Applied Sciences*, vol. 9, no. 19, p. 4062, 2019.

[4] A. Nikolov and V. Radivchev, "Nikolov-radivchev at semeval-2019 task 6: Offensive tweet classification with bert and ensembles," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, 2019, pp. 691–695.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[6] S. Garg and G. Ramakrishnan, "BAE: bert-based adversarial examples for text classification," *CoRR*, vol. abs/2004.01970, 2020.

[7] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack, "Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques," in *Third IEEE international conference on data mining*. IEEE, 2003, pp. 427–434.

[8] I. Pavlopoulos, "Aspect based sentiment analysis," *Athens University of Economics and Business*, 2014.

[9] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, "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*, Y. Bengio and Y. LeCun, Eds., 2015.

[11] L. Sun, K. Hashimoto, W. Yin, A. Asai, J. Li, P. Yu, and C. Xiong, "Adv-bert: Bert is not robust on misspellings! generating nature adversarial samples on bert," *arXiv preprint arXiv:2003.04985*, 2020.

[12] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is bert really robust? natural language attack on text classification and entailment," *arXiv preprint arXiv:1907.11932*, vol. 2, 2019.

[13] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," *arXiv preprint arXiv:1707.07328*, 2017.

[14] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 50–56.

[15] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "Hotflip: White-box adversarial examples for text classification," *arXiv preprint arXiv:1712.06751*, 2017.

[16] A. Karimi, L. Rossi, A. Prati, and K. Full, "Adversarial training for aspect-based sentiment analysis with bert," *arXiv preprint arXiv:2001.11316*, 2020.

[17] Y. Zang, F. Qi, C. Yang, Z. Liu, M. Zhang, Q. Liu, and M. Sun, "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*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 6066–6080.

[18] L. Li, R. Ma, Q. Guo, X. Xue, and X. Qiu, "Bert-attack: Adversarial attack against bert using bert," *arXiv preprint arXiv:2004.09984*, 2020.

[19] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang, "Generating natural language adversarial examples," *arXiv preprint arXiv:1804.07998*, 2018.

[20] A. Ek, J.-P. Bernardy, and S. Chatzikyriakidis, "How does punctuation affect neural models in natural language inference," in *Proceedings of the Probability and Meaning Conference (PaM 2020)*, 2020, pp. 109–116.

[21] R. He and J. McAuley, "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*, 2016, pp. 507–517.

[22] D. P. Kingma and J. Ba, "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*, Y. Bengio and Y. LeCun, Eds., 2015.

[23] A. Rietzler, S. Stabinger, P. Opitz, and S. Engl, "Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification," *arXiv preprint arXiv:1908.11860*, 2019.

[24] M. Zhang, J. Lucas, J. Ba, and G. E. Hinton, "Lookahead optimizer: k steps forward, 1 step back," in *Advances in Neural Information Processing Systems*, 2019, pp. 9597–9608.

[25] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "Semeval-2015 task 12: Aspect based sentiment analysis," in *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, 2015, pp. 486–495.

[26] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proceedings of the 26th International Conference on World Wide Web Companion*, 2017, pp. 759–760.

[27] Y. Zhu, R. Kiros, R. S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "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, 2015, pp. 19–27.