

# Model Extraction and Adversarial Transferability, Your BERT is Vulnerable!

Xuanli He<sup>†</sup>, Lingjuan Lyu<sup>‡</sup>, Qiongkai Xu<sup>\*</sup>, Lichao Sun<sup>§</sup>

<sup>†</sup> Monash University, xuanli.he1@monash.edu

<sup>‡</sup> Ant Group, lyulingjuan.llj@antgroup.com (Corresponding author)

<sup>\*</sup> The Australian National University and Data61 CSIRO, Qiongkai.Xu@anu.edu.au

<sup>§</sup> Lehigh University, lis221@lehigh.edu

## Abstract

Natural language processing (NLP) tasks, ranging from text classification to text generation, have been revolutionised by the pre-trained language models, such as BERT. This allows corporations to easily build powerful APIs by encapsulating fine-tuned BERT models for downstream tasks. However, when a fine-tuned BERT model is deployed as a service, it may suffer from different attacks launched by the malicious users. In this work, we first present how an adversary can steal a BERT-based API service (the victim/target model) on multiple benchmark datasets with limited prior knowledge and queries. We further show that the extracted model can lead to highly transferable adversarial attacks against the victim model. Our studies indicate that the potential vulnerabilities of BERT-based API services still hold, even when there is an architectural mismatch between the victim model and the attack model. Finally, we investigate two defence strategies to protect the victim model, and find that unless the performance of the victim model is sacrificed, both model extraction and adversarial transferability can effectively compromise the target models.

## 1 Introduction

Recently, owing to the success of pretrained BERT-based models (Devlin et al., 2018; Liu et al., 2019), the downstream NLP tasks have been revolutionised in the form of the limited task-specific supervision via fine-tuning on BERT models. Meanwhile, commercial task-oriented NLP models, built on top of BERT models, are often deployed as pay-per-query prediction APIs for the sake of the protection of data privacy, system integrity and intellectual property.

As publicly accessible services, commercial APIs have become victims of different explicit attacks, such as privacy attack (Lyu et al., 2020a,b; Shokri et al., 2017), adversarial attack (Shi et al.,

2018), etc. Recently, prior works have also found that with the aid of carefully-designed queries and outputs of the NLP APIs, many existing APIs can be locally imitated via model extraction (Krishna et al., 2019; Wallace et al., 2020), which raises concerns of the vulnerability of NLP APIs. For instance, competing companies can imitate the victim model with a negligible cost. Since the considerable investment of data annotation and algorithm design are sidestepped, the competing companies would be able to launch an identical service with a more competitive price than the victim companies. Such security issue can be exacerbated, when the back-end pertained models, such as BERT, are publicly available (Krishna et al., 2019).

Beyond model extraction, we further demonstrate the adversarial examples crafted by the extracted model could be transferred to the black-box victim model. From the perspective of commercial competition, if the competitors manage to predicate incorrect predictions of the victim services, they can launch an advertising campaign against the victim model with these adversarial examples.

In summary, we investigate the vulnerabilities of publicly available NLP classification APIs through a two-stage attack. First, a model extraction attack is issued to obtain a local copy of the target model. Then, we conduct adversarial attacks against the extracted model, which is empirically transferable to the target model. To patch these vulnerabilities, we mount two basic defence strategies on the victim models. The empirical results show that without corrupted predictions from the victims, model extraction and adversarial example transferability are resilient to the defence. Our results spotlight the risks of using pretrained BERT to deploy the APIs through the lens of model extraction attack and adversarial example transfer attack. Such attacks can be conducted at a cost of as little as \$7.1.<sup>1</sup>

<sup>1</sup>Code is available at [https://github.com/xlhex/extract\\_and\\_transfer](https://github.com/xlhex/extract_and_transfer)

## 2 Related Work

### 2.1 Model Extraction Attack (MEA)

Model extraction attacks (also referred to as “model stealing”) have been effectively applied to different tasks, ranging from computer vision tasks (Orekondy et al., 2019) to NLP tasks (Chandrasekaran et al., 2020).

In a nutshell, model extraction enables malicious users to forge the functionality of a black-box victim model as closely as possible. The activity seriously causes the intellectual property infringement. Additionally, the follow-up attacks can be facilitated as the aftermath of the model extraction. Particularly, an adversarial attack can be built upon the extracted model, which is able to enhance the successful rate of fooling the victim model.

### 2.2 Adversarial Transferability in NLP

As a byproduct of the adversarial attack, it has been shown that adversarial transferability encourages a transition of the adversarial examples from one model to other models (Liu et al., 2016; Papernot et al., 2017), especially in computer vision research. Although such property has been explored by a few recent works in NLP systems (Sun et al., 2020; Wallace et al., 2020), it remains largely unexplored for the BERT-based APIs, and whether the transferability could succeed when the substitute (extracted) model and the victim model have different architectures.

## 3 Attack on BERT-based API

Our attacks against BERT-based APIs consist of two phases, *Model Extraction Attack* (MEA) and *Adversarial Example Transfer* (AET), as depicted in Figure 1.

### 3.1 Model Extraction Attack (MEA)

In the first phase, we assume that a “victim model”  $M_v$  is commercially available as a prediction API for target task  $\mathcal{T}$ . An adversary attempts to reconstruct a local copy  $M_e$  (“extracted model”) of  $M_v$  via querying  $M_v$ . Our goal is to extract a model with comparable accuracy to the victim model. Generally, MEA can be formulated as a two-step approach, as illustrated by the left figure in Figure 1:

1. Attackers craft a set of inputs as queries, then send them to the victim model (BERT-based API) to obtain predictions;

Dataset	#Train	#Dev	#Test	Task
TP-US	22,142	2,767	2,767	sentiment analysis
Yelp	520K	40,000	1,000	sentiment analysis
AG	112K	1,457	1,457	topic classification
Blog	7,098	887	887	topic classification

Table 1: Statistic of sentiment analysis and topic classification datasets.

2. Attackers reconstruct a local copy of the victim model as an “extracted model” using the retrieved query-prediction pairs.

For each query  $x_i$ ,  $M_v$  returns a  $K$ -dim posterior probability vector  $y_i \in [0, 1]^k$ , with  $\sum_k y_i^k = 1$ . The resulting dataset  $\{x_i, y_i\}_{i=1}^m$  by  $m$  queries is used to train  $M_e$ . We assume that the attacker fine-tunes the public release of  $f_{\text{bert}, \theta^*}$  on this dataset, with the objective of imitating the behaviour of  $M_v$ . Once the local copy of  $M_e$  is obtained, the attacker no longer needs to pay the original service provider.

### 3.2 Adversarial Example Transfer (AET)

In the second phase, we leverage the transferability of adversarial examples: we first generate adversarial examples for the extracted model, then transfer the generated adversarial examples to the victim model. The intuition of the experiment is based on the transferable vulnerabilities crossing the models – the adversarial examples generated by the extracted model are transferable to the victim model. Here we use the extracted model to serve as a surrogate to craft adversarial examples in a white-box manner. Such attack aggravates the vulnerabilities of victim models.

## 4 Experiments and Analysis

### 4.1 NLP Tasks and Datasets

To evaluate the efficacy of the proposed attacks, we select four NLP datasets covering two main tasks, i) sentiment analysis and ii) topic classification. We use TP-US from Trustpilot Sentiment dataset (Hovy et al., 2015) and YELP dataset (Zhang et al., 2015) for sentiment analysis. We use AG news corpus (Del Corso et al., 2005) and Blog posts dataset from the blog authorship corpus (Schler et al., 2006) for topic classification. We refer readers to Appendix A for more details about the pre-processing of these datasets.

### 4.2 MEA Setup and Results

**Attack Strategies:** We assume that both victim and extracted models are initialised from a freely

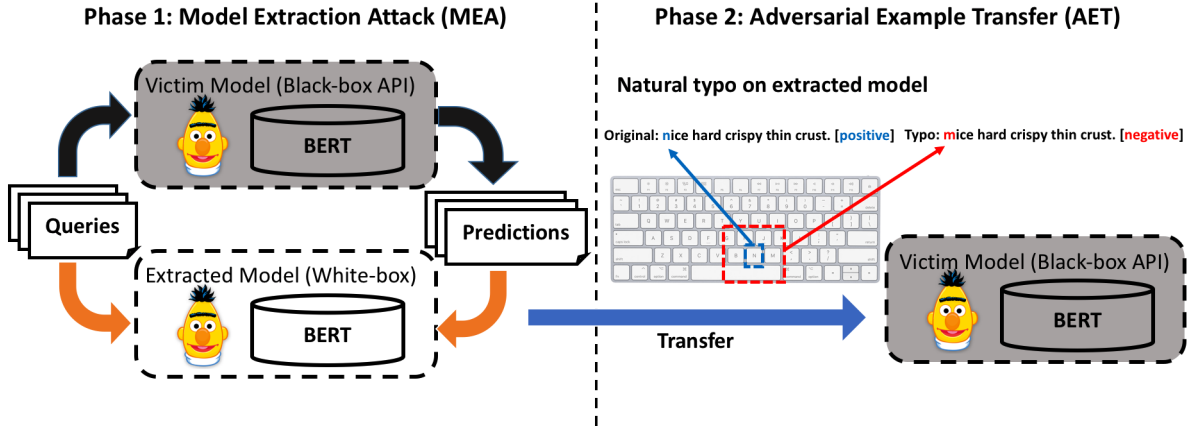


Figure 1: The workflow of the proposed attacks on BERT-based APIs. In phase 1, Model Extraction Attack (MEA) labels queries using the victim API, and then trains an extracted model on the resulting data. In phase 2, Adversarial Example Transfer (AET) generates adversarial typo examples on the extracted model, and transfers them to the victim API.

available pretrained BERT. Once the victim model is task-specifically fine-tuned by following Section 3.1, it can be queried as a black-box API. Afterwards, the extracted model can be obtained through imitating the victim model. Following Krishna et al. (2019), the queries start from the size of 1x to that of victim’s training set, then scale up to 5x. We test the accuracy of the victim model and the extracted model on the same held-out set for a fair comparison.

**Query Distribution:** To examine the correlation between the query distribution ( $D_A$ ) and the effectiveness of our attacks on the victim model trained on data from  $D_V$  (cf., Table 1), we explore the following two different scenarios: (1) we use the same data as the original data of the victim model ( $D_A = D_V$ ). Note that attackers have no true labels of the original data; (2) we sample queries from different distribution but same domain as the original data ( $D_A \neq D_V$ ).

Since the owners of APIs tend to use the in-house datasets, it is difficult for the attacker to know the target data distribution as a prior knowledge. Therefore, our second assumption is closer to the practical scenario. As the training datasets of the victims are sourced from either review domain or news domain, we consider datasets from these two domains as our queries. Specifically, we leverage Amazon review dataset (Zhang et al., 2015) or CNN/DailyMail dataset (Hermann et al., 2015) to query the victim models.

According to Table 2, we have observed that: 1) the success of the extraction correlates to the domain closeness between the victim’s training

Model	#Q	TP-US	Yelp	AG	Blog
Victim model		85.5	95.6	94.5	97.1
$D_A = D_V$		<b>86.5</b>	<b>95.7</b>	<b>94.5</b>	<b>96.8</b>
$D_A \neq D_V$ (review)	1x	85.3	94.1	88.6	88.2
	5x	85.8	95.0	91.3	92.8
$D_A \neq D_V$ (news)	1x	84.2	91.1	90.5	83.1
	5x	85.5	93.1	92.3	87.6

Table 2: Accuracy [%] of the victim models and the extracted models among different datasets in terms of domains and sizes. #Q: number of queries.

data and the attacker’s queries; 2) using same data even outperforms the victim models, which is also known as self-distillation (Furlanello et al., 2018); 3) albeit the different distributions brought by review and news corpora, our MEA can still achieve  $0.85\text{-}0.99\times$  victim models’ accuracies when the number of queries varies in  $\{1x, 5x\}$ . Although more queries suggest a better extraction performance, small query budgets (0.1x and 0.5x) are often sufficiently successful. More results are available in Appendix C. From now on, unless otherwise mentioned, we will use *news* data for AG news, and *review* data for TP-US, Blog and Yelp.<sup>2</sup>

**Costs Estimation:** We analyse the efficiency of MEA on various classification datasets. Each query is charged due to a pay-as-you-use policy adopted by service providers. We estimate costs for each task in Table 3 according to Google APIs<sup>3</sup> and IBM APIs<sup>4</sup>. Considering the efficacy of model extrac-

<sup>2</sup>Empirically, we do not have access to the original training data of the victim model.

<sup>3</sup><https://cloud.google.com/natural-language/pricing>

<sup>4</sup><https://www.ibm.com/cloud/watson-natural-language-understanding/>

Dataset	#Query	Google price	IBM price
TP-US	22,142	\$22.1	\$66.3
Yelp	520K	\$520.0	\$1,560.0
AG	112K	\$112.0	\$336.0
Blog	7,098	\$7.1	\$21.3

Table 3: Estimate costs of model extraction on different datasets

tion, the cost is highly economical and worthwhile.

### 4.3 AET Setup and Results

After extracting a black-box victim model into a white-box extracted model, a white-box adversarial attack can be implemented. We first generate adversarial examples on the extracted model, then examine whether these examples are transferable to the target victim model. To evaluate such pseudo white-box attack, we assess it via a transferability metric, which refers to the misclassification rate of adversarial samples on the victim APIs.

To generate natural adversarial examples, we follow the protocol (Sun et al., 2020) that leverages the gradients of the gold labels w.r.t the embeddings of the input tokens to find the most informative tokens, which have the largest gradients among all positions within a sentence. Then we corrupt the selected tokens with one of the following typos: 1) Insertion; 2) Deletion; 3) Swap; 4) Mistype: Mistyping a word though keyboard, such as “oh” → “0h”; 5) Pronounce: Wrongly typing due to the close pronounce of the word, such as “egg” → “agg”; 6) Replace-W: Replace the word by the frequent human behavioural keyboard typo based on the Wikipedia statistics (Sun, 2020).

In order to understand whether our extracted model manages to improve the transferability, we also launch a list of black-box adversarial attacks in the same manner. Table 4 demonstrates that our pseudo white-box attack makes the victim model more vulnerable to adversarial examples in terms of transferability — more than twice effective in the best case, compared to the black-box counterparts. This corroborates our claim that the extracted model, retaining a high-fidelity imitation of the victim model, severely impairs the output integrity of the victim model, indicated as the considerable increase of the transferable examples.

In general, Table 4 also shows that more queries

pricing

		TP-US	Yelp	AG	Blog
black-box	deepwordbug				
	1x	18.4	18.5	25.6	52.9
	5x	18.2	25.7	35.3	67.8
	textbugger				
	1x	21.3	16.3	16.1	41.2
	5x	21.1	21.3	24.7	62.7
	textfooler				
	1x	27.5	17.3	18.5	34.7
	5x	27.1	21.9	24.9	64.4
	w-box (ours)	adv-bert			
	1x	<b>48.6</b>	35.5	47.5	64.9
	5x	47.3	<b>43.3</b>	<b>53.6</b>	<b>76.5</b>

Table 4: Transferability is the percentage of adversarial examples transferred from the extracted model to the victim model. deepwordbug (Gao et al., 2018); textbugger (Li et al., 2018); textfooler (Jin et al., 2019); advbert (Sun et al., 2020). w-box: white-box.

Victim	Extracted	MEA	AET
BERT-large	BERT-large	<b>91.0</b>	<b>59.3</b>
BERT-base	BERT-large	90.7	37.2
BERT-base	BERT-base	90.5	47.5
BERT-large	BERT-base	89.9	42.7

Table 5: Attack performance on AG news with mismatched BERT architectures.

(5x v.s. 1x) lead to better attack performances. We believe this conspicuous gain attributes to the higher fidelity to the victim model, obtained by a better extraction (*c.f.*, Table 2).

### 4.4 Architecture Mismatch

In practice, the adversary may not know the victim’s model architecture. Hence we also study the attacking behaviours under the different architectural settings. According to Table 5, when both the victim and the extracted models adopt BERT-large, the vulnerability of the victim is magnified in all attacks, which implies that the model with higher capability is more vulnerable to our attacks. As expected, the efficacy of AET can be alleviated when an architectural mismatch exists.<sup>5</sup>

## 5 Defence

We next briefly discuss two defence strategies the victim model can adopt to counter these attacks.

- **Softening predictions (SOFT).** A temperature coefficient  $\tau$  on softmax layer manipulates the posterior probability distribution.

<sup>5</sup>More experiments can be found in Appendix D

	TP-US		Yelp		AG		Blog	
	MEA ↓	AET ↓	MEA ↓	AET ↓	MEA ↓	AET ↓	MEA ↓	AET ↓
NO DEF.	85.3 (85.5)	48.6	94.1 (95.6)	35.5	90.5 (94.5)	47.5	88.2 (97.1)	64.9
SOFT. ( $\tau=0.0$ )	84.6 (85.5)	40.2	93.7 (95.6)	21.6	90.0 (94.5)	33.0	85.6 (97.1)	51.4
SOFT. ( $\tau=0.5$ )	85.1 (85.5)	50.9	93.8 (95.6)	20.6	90.3 (94.5)	33.1	85.7 (97.1)	61.5
SOFT. ( $\tau=5.0$ )	85.3 (85.5)	58.7	94.5 (95.6)	36.1	90.9 (94.5)	53.3	86.7 (97.1)	66.7
PERT. ( $\sigma=0.05$ )	85.3 (85.5)	55.0	93.9 (95.6)	29.2	90.1 (94.3)	40.3	85.9 (96.2)	64.0
PERT. ( $\sigma=0.20$ )	85.1 (85.4)	49.7	93.7 (95.5)	25.4	90.2 (94.3)	35.4	85.3 (95.4)	52.2
PERT. ( $\sigma=0.50$ )	<b>82.7</b> (63.2)	<b>28.3</b>	<b>92.5</b> (87.8)	<b>16.6</b>	<b>89.0</b> (76.4)	<b>20.0</b>	<b>81.8</b> (62.2)	<b>32.8</b>

Table 6: Attack performance under different defences (NO DEF., SOFT. and PERT.) and datasets. **Lower** scores indicate better defences. All experiments are conducted with 1x queries. Numbers in parentheses are accuracy of victim models with defence.

A higher  $\tau$  leads to smoother probability, whereas a lower one produces a sharper distribution. When  $\tau=0$ , the posterior probability becomes a hard label.

- **Prediction perturbation (PERT).** Another defence method is adding normal noise with variance  $\sigma$  to the predicted probability distribution. The larger the variance of the noise distribution, the stronger the defence.

Table 6 indicates that varying temperature on softmax cannot defend the victim model against MEA, except for  $\tau=0$  (hard label), which can degrade all attacks to some extent.

Regarding perturbation, it can achieve a significant defence at the cost of the accuracy of the victim models. Surprisingly, when  $\sigma=0.50$ , MEA surpasses the victim model. We conjecture that albeit the perturbed post-softmax probability, the extracted model can still acquire certain informative knowledge via model extraction. We will conduct an in-depth study on this in the future.

To sum up, both MEA and AET pose severe threats to the BERT-based APIs, even when the adversary merely has access to limited or erroneous predictions.

## 6 Conclusions

This work goes beyond model extraction from BERT-based APIs, and we also identify the extracted model can largely enhance adversarial example transferability even in difficult scenarios, *i.e.*, limited query budget, queries from different distributions, or architectural mismatch. Extensive experiments based on representative NLP datasets and tasks under various settings demonstrate the effectiveness of our attacks against BERT-based APIs. In the future, we plan to extend our work

to more complex NLP tasks, and develop more effective defences.

## Acknowledgements

We would like to thank anonymous reviewers for their valuable feedback and constructive suggestions. The computational resources of this work are supported by the Multi-modal Australian ScienceS Imaging and Visualisation Environment (MASSIVE) ([www.massive.org.au](http://www.massive.org.au)).

## References

- Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. 2020. Exploring connections between active learning and model extraction. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 1309–1326.
- Maximin Coavoux, Shashi Narayan, and Shay B. Cohen. 2018. Privacy-preserving neural representations of text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1–10.
- Gianna M Del Corso, Antonio Gulli, and Francesco Romani. 2005. Ranking a stream of news. In *Proceedings of the 14th international conference on World Wide Web*, pages 97–106.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born again neural networks. *arXiv preprint arXiv:1805.04770*.
- 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. IEEE.

- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.
- Dirk Hovy, Anders Johannsen, and Anders Søgaard. 2015. User review sites as a resource for large-scale sociolinguistic studies. In *Proceedings of WWW*, pages 452–461.
- Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *arXiv*, pages arXiv–1907.
- Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. 2019. Thieves on sesame street! model extraction of bert-based apis. *arXiv preprint arXiv:1910.12366*.
- 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*.
- Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2016. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Ilya Loshchilov and Frank Hutter. 2019. **Decoupled weight decay regularization**. In *International Conference on Learning Representations*.
- Lingjuan Lyu, Xuanli He, and Yitong Li. 2020a. Differentially private representation for nlp: Formal guarantee and an empirical study on privacy and fairness. *arXiv preprint arXiv:2010.01285*.
- Lingjuan Lyu, Yitong Li, Xuanli He, and Tong Xiao. 2020b. Towards differentially private text representations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1813–1816.
- Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4954–4963.
- 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*, pages 506–519.
- Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205.
- Yi Shi, Yalin E Sagduyu, Kemal Davaslioglu, and Jason H Li. 2018. Generative adversarial networks for black-box api attacks with limited training data. In *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 453–458. IEEE.
- R. Shokri, M. Stronati, C. Song, and V. Shmatikov. 2017. **Membership inference attacks against machine learning models**. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18.
- Lichao Sun. 2020. Natural backdoor attack on text data. *arXiv preprint arXiv:2006.16176*.
- 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*.
- Eric Wallace, Mitchell Stern, and Dawn Song. 2020. Imitation attacks and defenses for black-box machine translation systems. *arXiv preprint arXiv:2004.15015*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. **Transformers: State-of-the-art natural language processing**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

## A Dataset Description

**Trustpilot (TP).** Trustpilot Sentiment dataset (Hovy et al., 2015) contains reviews associated with a sentiment score on a five point scale. The original dataset is comprised of reviews from different locations, however in this paper, we only derive TP-US for study.

**AG news.** We use AG news corpus (Del Corso et al., 2005). This task is to predict the topic label of the document, with four different topics in total. Following (Zhang et al., 2015; Jin et al., 2019), we use both “title” and “description” fields as the input document.

**Blog posts (Blog).** We derive a blog posts dataset (Blog) from the blog authorship corpus presented (Schler et al., 2006). We recycle the corpus preprocessed by Coavoux et al. (2018), which covers 10 different topics.

**Yelp Polarity (Yelp).** Yelp dataset is a document-level sentiment classification (Zhang et al., 2015). The original dataset is in a five point scale (1-5), while the polarised version assigns negative labels to the rating of 1 and 2 and assigns positive labels to 4 and 5.

## B Training Details

We use Huggingface (Wolf et al., 2020) as the codebase. Each model is trained for 4 epochs on a NVIDIA V100 GPU, with a batch size of 64. We use AdamW (Loshchilov and Hutter, 2019) with a learning rate of  $5e-5$ .

## C Performance of Different Query Size

Due to the budget limit, the attacker cannot issue massive requests. To investigate the attack performance of model extraction under the low-resource setting, we conduct two additional experiments, which only utilise 0.1x and 0.5x of the training data of the victim models respectively. According to Table 7, the overall performance of extracted models is comparable to the victim models. Only Blog with 0.1x training suffers from a drastic drop, as Blog uses the least number of training samples in all four datasets. In addition, distant domains exhibit significant degradation, when compared to the close ones. For example, sampling 0.1x-5x queries from news data present a more stable attack performance against the victim model trained with AG news than Blog.

	#Q	AG	Blog	TP-US	Yelp
victim model		94.47	97.07	85.53	95.57
$D_A = D_V$		94.54	96.77	86.48	95.72
$D_A \neq D_V$ (review)	0.1x	86.57	36.83	79.95	92.39
	0.5x	87.31	84.59	84.21	93.25
	1x	88.63	88.16	85.33	94.06
	5x	91.27	92.75	85.82	94.95
$D_A \neq D_V$ (news)	0.1x	89.13	18.04	79.20	88.24
	0.5x	89.84	32.92	84.18	89.76
	1x	90.48	83.13	84.15	91.06
	5x	92.26	87.64	85.46	93.13

Table 7: Accuracy [%] of the victim models and the extracted models among different datasets in terms of domains and sizes. #Q: number of queries.

Victim	Extracted	MEA	AET
BERT-large	BERT-base	89.88	42.7
RoBERTa-large	BERT-base	89.74	27.7
RoBERTa-base	BERT-base	89.45	36.4
XLNET-large	BERT-base	89.66	32.7
XLNET-base	BERT-base	89.27	34.4
BERT-base	BERT-base	90.48	47.5

Table 8: Attack performance on AG news with mismatched BERT architectures.

## D Architectural Mismatch

In Table 8, we experiment with different models, including BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019) and XLNET (Yang et al., 2019). Although the architectural difference can cause some drops in MEA and AET, overall the proposed attacks are still effective.