Layerwise universal adversarial attack on NLP models

Olga Tsymboi[1, 2], **Danil Malaev**[1], **Andrei Petrovskii**[1], and **Ivan Oseledets**[3, 4]

[1]Sber AI Lab, Moscow, Russia
[2]Moscow Institute of Physics and Technology, Moscow, Russia
[3]Skolkovo Institute of Science and Technology, Moscow, Russia
[4]Artificial Intelligence Research Institute (AIRI), Moscow, Russia
tsimboy.oa@phystech.edu, m9255171362@gmail.com, andreypetrovskij@gmail.com
i.oseledets@skoltech.ru

Abstract

In this work, we examine the vulnerability of language models to universal adversarial triggers (UATs). We propose a new white-box approach to the construction of *layerwise* UATs (LUATs), which searches the triggers by perturbing hidden layers of a network. On the example of three transformer models and three datasets from the GLUE benchmark, we demonstrate that our method provides better transferability in a model-to-model setting with an average gain of 9.3% in the fooling rate over the baseline. Moreover, we investigate triggers transferability in the task-to-task setting. Using small subsets from the datasets similar to the target tasks for choosing a perturbed layer, we show that LUATs are more efficient than vanilla UATs by 7.1% in the fooling rate.

1 Introduction

One of the fundamental drawbacks of modern neural networks is their vulnerability to adversarial attacks (Szegedy et al., 2013; Goodfellow et al., 2014), imperceptible perturbations to the data samples that leave the ground truth label unchanged but are able to modify model prediction drastically. The samples obtained as the result of these perturbations are called adversarial examples. First discovered for image datasets (Szegedy et al., 2013), this phenomenon was then demonstrated for other types of data, including natural language (Papernot et al., 2016; Liang et al., 2017; Gao et al., 2018).

The originally proposed methods of adversarial attack construction were sample-dependent, which means that one can not apply the same perturbation to different dataset items and expect equal success. Sample-agnostic, or in other words, universal, adversarial perturbations (UAPs) were proposed in Moosavi-Dezfooli et al. (2017), where based on a small subset of image data, the authors constructed perturbations leading to prediction change of 80-90% (depending on the model) of samples. They also showed that UAPs discovered on one model could successfully fool another.

The generalization of the universal attacks to natural language data was made by Wallace et al. (2019). Short additives (triggers) were inserted at the beginning of data samples, and then a search over the token space, in order to maximize the probability of the negative class on the chosen data subset, was performed. The found triggers turned out to be very efficient at fooling the model, repeating the success of UAPs proposed for images.

The conventional way to look for adversarial examples is to perturb the output of a model. Considering image classification neural networks, Khrulkov and Oseledets (2018) proposed to search for perturbations to hidden layers by approximating the so-called (p, q)-singular vectors (Boyd, 1974) of the corresponding Jacobian matrix. Then one can hope that the error will propagate through the whole network end, resulting in model prediction change. They showed that this approach allows obtaining a high fooling rate based on significantly smaller data subsets than those leveraged by Moosavi-Dezfooli et al. (2017).

In this paper, we aim to continue the investigation of neural networks' vulnerability to universal adversarial attacks in the case of natural language data. Inspired by the approach considered by Khrulkov and Oseledets (2018), we look for the perturbations to hidden layers of a model instead of the loss function. In order to avoid projection from embedding to discrete space, we use simplex parametrization of the search space (see, e.g. (Dong et al., 2021; Guo et al., 2021)). We formulate the corresponding optimization problem and propose the algorithm for obtaining its approximate solution. On the example of three transformer models: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2020) and three GLUE datasets (Wang et al., 2018) we demonstrate higher efficiency of our method over the original approach

Findings of the Association for Computational Linguistics: ACL 2023, pages 129–143
July 9-14, 2023 ©2023 Association for Computational Linguistics

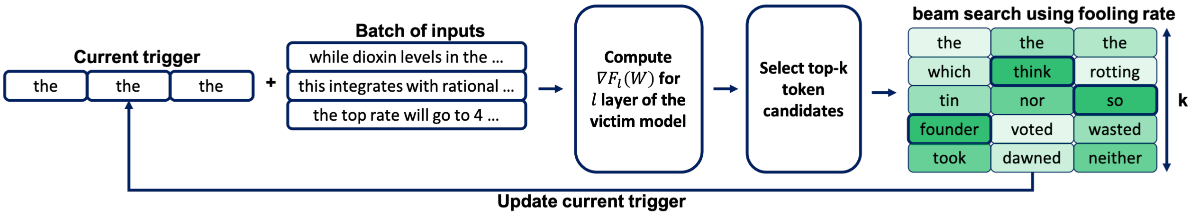

Current trigger
the
the
the
+
Batch of inputs
while dioxin levels in the ...
this integrates with rational ...
the top rate will go to 4 ...
Compute $\nabla F_l(W)$ for l layer of the victim model
Select top-k token candidates
beam search using fooling rate
the
the
the
which
think
rotting
tin
nor
so
founder
voted
wasted
took
dawned
neither
k
Update current trigger

Figure 1: LUATs (ours): Layerwise Universal Adversarial Triggers. The algorithm scheme, where ∇F_l is defined in (8).

of Wallace et al. (2019) in the setting of model-to-model and task-to-task transfer. We also show that in the case of direct attack application, our method demonstrates the results which are on par with the baseline, where perturbation of the loss function was realized. We hope that this technique will serve as a useful tool for discovering flaws and deepening our understanding of language neural networks.

2 Framework

Let $f : \mathcal{X} \to \mathcal{Y}$ be a text classification model defined for a dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, where x_i is an input and y_i is a corresponding label. Our goal is to find a small perturbation to the original input sequences that leads to a change in a model prediction for the maximum possible number of samples. Such perturbation could be modelled as an insertion of a trigger, a token sequence t of small length L, into a particular part of an input. Following Wallace et al. (2019), we call such text perturbations universal adversarial triggers (UATs). In this paper, we focus on triggers concatenated to the front of a sentence and denote a corrupted sample as $\hat{x} = t \oplus x$. It is also important to note that, in contrast to Wallace et al. (2019), we consider the unsupervised scenario when an attacker does not have access to the labels. For evaluation we follow Moosavi-Dezfooli et al. (2017) and use the *fooling rate* (FR):

$$FR = \frac{1}{N} \sum_{\hat{x}=t\oplus x,\ x\in\mathcal{D}} [f(x) \neq f(\hat{x})]. \qquad (1)$$

Universal adversarial triggers. Before we dive into the description of our approach, it is worth depicting the original method of UATs. Restricting themselves to the white-box setting, Wallace et al. (2019) showed that an efficient way to find UATs could be performed through the optimization of the first-order expected loss approximation:

$$\max_{t\in\mathcal{V}^L} \mathbb{E}_{x,y\sim\mu} \langle \nabla_{t^n} \mathcal{L}(t^n \oplus x, y), t - t^n \rangle, \qquad (2)$$

where μ is a distribution of input data, t^n denotes the trigger found after the n-th iteration, $\mathcal{V}$ stands for the token vocabulary and the initial trigger t^0 can be chosen as the L-time repetition of the word "the". For simplicity, here (and below in similar situations), by t, we mean its embedding if the gradient is taken with respect to it or if it appears as a term in a scalar product.

In order to find the optimal perturbation within each iteration, the expectation in (2) is relaxed with the average over a batch, and maximization is performed for each trigger token independently:

$$\max_{t_j\in\mathcal{V}} \sum_{x,y\in\text{batch}} \langle \nabla_{t_j^n} \mathcal{L}(t^n \oplus x, y), t_j - t_j^n \rangle, \qquad (3)$$

where j is the token index. After solving (3), the next trigger t^{n+1} is selected via beam search over all the token positions.

Our approach: layerwise universal adversarial triggers (LUATs). In (3), we face the optimization problem over a discrete set of tokens from the vocabulary $\mathcal{V}$. To overcome this issue, let us relax (3) using the probability simplex model for every token. In this case, a trigger can be represented as $t = WV$, where V is a vocabulary matrix and W_{mn} is a probability of the n-th token from vocabulary to be selected at position m. Then (3) can be rewritten as follows:

$$\max_{W\in\mathcal{S}} \sum_{x,y\in\text{batch}} \langle \nabla_{t^n} \mathcal{L}(t^n \oplus x, y), WV - t^n \rangle, \qquad (4)$$

where $\mathcal{S} = \{W |\ W\mathbb{1} = 1,\ W \geq 0\}$ and $W \geq 0$ denotes an element-wise inequality. In this formulation, the search for the solution is done by performing optimization of the weights W over the simplex $\mathcal{S}$.

Our approach can be seen as an extension of (4) to the perturbation of hidden layers. This is inspired by Khrulkov and Oseledets (2018), who applied this idea to find UAPs for fooling image classification neural networks. Given a layer l, the optimization problem, in this case, can be similarly obtained via the Taylor expansion:

$$\begin{gathered} l(\hat{x}) - l(\hat{x}^n) \approx J_l(\hat{x}^n)(\hat{x} - \hat{x}^n) \\ \|l(\hat{x}) - l(\hat{x}^n)\|_q^q \to \max_{t \in \mathcal{V}^L}, \end{gathered} \tag{5}$$

where $\hat{x}^n = t^n \oplus x$, q is a hyperparameter to be fine-tuned and $J_l(x)$ is the Jacobian operator corresponding to a layer l:

$$J_l(x) = \frac{\partial l(x)}{\partial x}. \tag{6}$$

Bringing (4) and (5) together we obtain

$$\begin{aligned} &\max_{W \in \mathcal{S}} F_l(W) = \\ &\max_{W \in \mathcal{S}} \sum_{x \in \text{batch}} \|J_l(t^n \oplus x)(WV - t^n)\|_q^q. \end{aligned} \tag{7}$$

In contrast to (3), finding the optimal solution of (7) is computationally infeasible. Indeed, the problem (3) allows a brute-force approach for finding the optimal token for each trigger position since it requires computing the gradient only once for each iteration. On the other hand, in our case, a brute-force computation is very cumbersome since, for each iteration, it would require computing the Jacobian action for every batch, token candidate and position, resulting in $\mathcal{O}(LB|\mathcal{V}|)$ forward-backward passes, where B is the number of batches in an iteration. Luckily, $F_l(W)$ is convex and can be lower bounded via a tangent line, where the gradient is calculated as follows:

$$\begin{aligned} &\nabla F_l(W) = \\ &\sum_{x \in \text{batch}} J_l^\top(\hat{x}^n)\psi_q(J_l(\hat{x}^n)(WV - t^n))V^\top, \end{aligned} \tag{8}$$

where $\psi_q(x) = \text{sign}(x)|x|^{q-1}$. Therefore, our task is reduced to finding the solution to the linear problem with the simplicial constraint:

$$\max_{W \in \mathcal{S}} \langle \nabla F_l(W^*), W \rangle, \tag{9}$$

where $\nabla F_l(W)$ is given by (8) and W^* denotes the point where we perform the linear approximation. The final problem (9) has a closed-form solution (see the Appendix A for more details) and, as a result, we reduced the number of forward-backward computations to $\mathcal{O}(B)$.

Concerning the initialization of W^*, we take the uniform distribution over all the vocabulary tokens for each token position in a trigger. Within each iteration, we perform only one step with respect to W in order to reduce computation time and observe that it is sufficient for breaking the models efficiently.

Finally, since the found after a given iteration weight matrix W in the worst case has only one non-zero element per row (see Appendix A), we can get into a local maximum unless we guess the proper initialization. Therefore, similarly to Wallace et al. (2019), we perform a beam search over the top-k candidates. In order to realize it, it is necessary to define the ranking criterion for choosing the best option at each search step. For this purpose, we use the FR. The overall algorithm is presented in Algorithm 1 and Fig. 1.

Algorithm 1: LUATs: Layerwise Universal Adversarial Triggers

```
Input: Dataset $\mathcal{D}$, victim model, tokenizer,
       $q$, layer to attack $l$, trigger length $L$,
       top-$k$, beam size $b$
Output: Trigger $t$
$t$ = tokenizer($\underbrace{\text{the} \dots \text{the}}_{L\ times}$)
$W = \frac{1}{|\mathcal{V}|}\text{ones}(L, |\mathcal{V}|)$
while FR increase do
    Sample batch $X$
    Compute $\nabla F_l(W)$ over batch
    candidates = Select indices of $k$
      largest entries of $\nabla F_l(W)$ for each
      token position
    $t$ = BeamSearch($\mathcal{D}$, candidates, $b$) to
      maximize FR
```

3 Experiments

In this section, we present a numerical study of the proposed layerwise adversarial attack framework on text classification datasets. The code is publicly available on GitHub[1].

3.1 Setup

Datasets. As in the work of Wang et al. (2021), we consider only a subset of tasks from the GLUE

[1] https://github.com/sb-ai-lab/nlp-layerwise-fooler

benchmark. In particular, we use **SST-2** for the sentiment classification, **MNLI** (matched), **QNLI**, **RTE** for the natural language inference and **MRPC** as the paraphrase identification. We exclude **CoLA**, which task is to define whether the input is grammatically correct or not, and universal triggers are highly probable to change most of the ground-truth positive labels to negative. Finally, **WNLI** is not considered because it contains too few examples. We conduct our experiments using the validation set for attack fitting and the test set for evaluation.

Models. We focus our consideration on three transformer models: BERT base, RoBERTa base and ALBERT base using the pre-trained weights from the TextAttack project (Morris et al., 2020) for most of the model-dataset cases. For some of them, the performance was unsatisfactory (all the models on MNLI, ALBERT on QNLI, RoBERTa on SST-2, RTE), and we fine-tuned them on the corresponding training sets (Mosin et al., 2023). To train our attack, we use existing GLUE datasets splits. The detailed statistics is presented in Tab. 1

Hyperparameters. In our experiments, we investigate the attack performance depending on the dataset, model, layer l (from 0 to 11), trigger length $L \in \{1,2,3,4,5,6\}$ and $q \in \{2,3,4,5,7,10\}$. For each dataset, model and trigger length, we performed a grid search over l and q. The other parameters, such as top-k and the beam size, remain fixed to the values obtained from the corresponding ablation study. In all experiments, we use a batch size of 128. Finally, we define the initialization of W as the uniform distribution over the vocabulary tokens for each position in a trigger (see Algorithm 1).

Token filtration and resegmentation. Transformers' vocabulary contains items such as symbols, special tokens and unused words, which are easily detected during the inference. To increase triggers' imperceptibility, we exclude them from the vocabulary matrix during optimization leaving only those which contain english letters or numbers.

Another problem appears since a lot of tokens do not correspond to complete words but rather pieces of words (sub-words). As a result, if the first found token corresponds to a sub-word, one encounters the retokenization, meaning that, after converting the found trigger to string and back, the set of the tokens can change. Moreover, sometimes one has to deal with appearing symbols such as "##" in a trigger. In this case, we drop all the extra symbols and perform the retokenization. Luckily, it does not result in severe performance degradation. In the case when, due to the resegmentation, the length of a trigger changes, we report the result as for the length for which the attack training was performed. As an alternative to direct resegmentation, we tried to transform triggers by passing them through an MLM model, but this approach led to a more significant drop in performance.

3.2 Main Results

Comparison with the baseline. We perform a comparison of LUATs with untargeted UATs of Wallace et al. (2019). In order to stay in the unsupervised setting, we modify their approach by replacing the ground truth labels in the cross-entropy loss function with the class probabilities. As a result, we search for a trigger that maximizes the distance between model output distributions before and after the perturbation. In addition, as the criterion for choosing the best alternative in the beam search, we use FR for both methods.

We perform the ablation study to estimate the dependence of FR on top-k and the beam size. For the beam size 1, we measure both attacks' performance for different values of top-k from 1 to 40. Then for the best top-k, we build the dependence on the beam size from 1 to 5. We perform this study on the QNLI dataset. The results are presented in the Fig. 5. We stick to the top-k 10 and the beam size 1 as a trade-off between high performance and low computational complexity.

The grid search results are presented in Tab. 2, where for each model, dataset, and trigger length, we show the best results of both approaches. We performed the computation on four GPU's NVIDIA A100 of 80GB. To reduce the influence hyperparameters searching space cardinality (72 times more runs due to different values of q and L), in Fig. 2, we present a time, averaged over 10 batches, per one iteration of both approaches. Indeed, for LUATs, we observe linear dependence on a layer, particularly 7.27 seconds on average versus 8.05 seconds for UATs of Wallace et al. (2019). The triggers obtained with the method of Wallace et al. (2019) took only 5 GPU hours; hence the execution time of the full grid search for LUATs could be estimated by 325 GPU hours. While the proposed method might be more efficient on average, high variance is explained by significantly different paddings in sampled batches.

Dataset	Validation	Test	# Classes	BERT Acc. Val. / Test	RoBERTa Acc. Val. / Test	ALBERT Acc. Val. / Test
MRPC	408	1725	2	87.7/84.4	90.4/87.2	89.7/86.0
QNLI	5463	5463	2	91.5/90.7	91.8/91.8	90.6/90.8
MNLI	9815	9796	3	84.2/83.7	86.5/86.3	83.8/83.5
SST-2	872	1821	2	92.4/93.3	94.0/94.9	92.7/91.7
RTE	277	3000	2	72.6/67.6	80.5/74.0	76.0/72.2

Table 1: Statistics of the considered classification datasets. We present datasets cardinality, the number of classes and model accuracy on both validation and test sets.

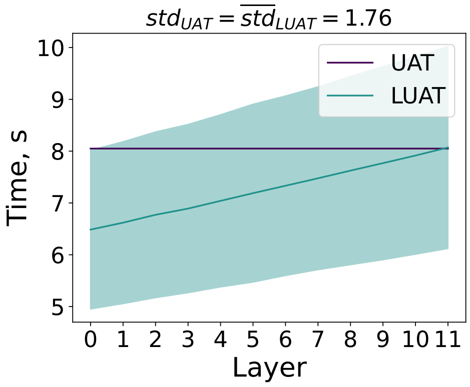


Figure 2: The iteration average time for UAT and LUAT (ours) approaches. The results presented on the QNLI dataset with fixed $q = 10$, $L = 3$ and a batch size of 128. Since, for the case of LUATs, the standard deviation does not change significantly, we report the value averaged over layers as $\overline{std}_{LUAT}$.

It is interesting to note that in some cases, shorter triggers appear to be better than longer ones (see Appendix A, Tab. 10). That is why we present the best performance for each length L over all the lengths less or equal to L. From Tab. 2, one can see that our method demonstrates the results, which are on par with Wallace et al. (2019). In Fig. 3, we present the dependence of FR on the trigger length L. The attack performance saturates when it approaches 5 or 6, meaning that considering longer triggers would hardly bring any performance gain.

Wallace et al. (2019) suggested that the efficiency of universal adversarial triggers can be explained by the existence of dataset biases, such as fallacious correlations between certain words and classes and supported this conjecture by computing pointwise mutual information (PMI) of vocabulary tokens and dataset classes

$$\mathrm{PMI}(token, class) = \log \frac{p(token, class)}{p(token)p(class)}.$$

As a result, they reported a high PMI rank for their tiggers tokens. In order to verify whether LUATs satisfy the same pattern, we perform this computation for our best triggers. We sample 5 best candidates obtained during the grid search for each trigger length and compute PMI rank with add-100 smoothing for their tokens. The results on QNLI[2] (see the Tab. 3) demonstrate that similarly to UATs our trigger tokens have high PMI ranks.

Dependence on q and a layer. For the investigation of dependence on q and a layer, we restrict ourselves to the datasets which appear to be the most vulnerable in our experiments: MRPC, QNLI and SST. The results are presented in Fig. 4, where we performed averaging over the lengths. One can conclude that, in general, it is more efficient to attack the higher layers. This observation can be interpreted with the idea which has been mentioned above; namely, that the efficiency of the triggers is caused by the existence of dataset biases. Indeed, as Merchant et al. (2020) demonstrated, fine-tuning for a downstream task primarily affects the higher layers of a model. Therefore, the bias which could be acquired due to fine-tuning should be accumulated in its higher layers. Since we try to fool a model with respect to a downstream task, the appearance of the higher layers among the most successful ones is more probable. Finally, it is interesting to note that the dependence on q also demonstrates better results for the larger values, which is in accordance with the findings of Khrulkov and Oseledets (2018).

Transferability: model-to-model. It was demonstrated that the universal triggers could be transferable between different models trained on the same task (Wallace et al., 2019). Here, we perform a comparison of their approach and ours with respect to this property. Similarly to the above consideration, the computations are carried out on MRPC, QNLI and SST-2 datasets, with an additional restriction on the trigger length $L = 3$, which

[2] The results on SST-2 and MRPC are presented in the Appendix B on the Tab. 6 and 7

Model	Dataset	FR@1		FR@2		FR@3		FR@4		FR@5		FR@6	
		W	S	W	S	W	S	W	S	W	S	W	S
ALBERT	MNLI	**18.5**	14.8	**25.8**	17.0	**25.8**	18.5	**34.8**	21.1	**34.8**	24.8	**34.8**	24.8
	MRPC	**11.4**	4.5	27.6	**28.9**	47.9	**54.8**	**66.6**	64.1	66.6	67.0	67.6	67.6
	QNLI	26.4	**37.5**	47.5	**48.1**	50.5	50.8	50.5	**51.5**	52.2	52.4	52.5	52.4
	RTE	4.8	4.9	6.3	**11.9**	7.9	**11.9**	9.1	**17.7**	9.1	**17.7**	9.1	**17.7**
	SST-2	21.0	**29.7**	34.8	**38.8**	45.5	**46.4**	47.9	**48.5**	**50.6**	48.7	**50.6**	48.7
BERT	MNLI	**32.0**	10.2	**38.4**	33.6	**38.4**	34.5	**38.4**	34.7	**38.4**	35.6	**38.4**	35.6
	MRPC	**15.4**	9.3	**64.6**	56.6	**70.6**	69.0	70.7	70.4	70.8	70.6	70.8	70.7
	QNLI	**26.2**	23.2	**42.6**	37.9	**47.2**	42.3	**50.6**	49.1	50.6	50.8	50.8	50.8
	RTE	**4.7**	4.2	6.6	6.7	7.1	**8.5**	7.6	**10.2**	7.6	**10.2**	**11.2**	10.2
	SST-2	**29.7**	27.0	**42.1**	38.0	42.1	**44.6**	42.1	**48.2**	**50.6**	49.0	**50.8**	49
RoBERTa	MNLI	4.8	**8.2**	4.9	**22.3**	**33.0**	28.8	**33.0**	30.0	33.0	33.1	33.0	33.1
	MRPC	4.2	**4.9**	**35.5**	26.3	67.8	**68.5**	69.4	69.3	69.4	69.3	69.5	69.3
	QNLI	**14.7**	13.3	32.6	**39.0**	41.5	**44.9**	44.5	**47.3**	48.1	47.7	48.1	48.3
	RTE	3.1	**6.3**	6.1	**13.6**	6.1	**13.6**	8.0	**13.6**	8.0	**16.3**	9.2	**16.3**
	SST-2	**28.4**	26.9	34.7	**38.7**	43.9	**45.9**	**49.9**	47.1	49.9	**50.7**	49.9	**53.1**
Average	-	**16.3**	15.0	30.0	**30.5**	38.4	**38.9**	41.5	41.5	42.7	42.9	43.1	43.2

Table 2: Comparison between LUAT (S) and UATs of Wallace et al. (2019) (W). We report the best FR on the test sets for triggers whose length does not exceed L (FR@L).

BERT				ALBERT				RoBERTa			
E	Rank	N	Rank	E	Rank	N	Rank	E	Rank	N	Rank
emperor	94.69	either	99.60	70%	98.95	those	99.02	billion	99.95	why	99.65
berlin	93.72	legislation	97.60	10%	96.10	amount	98.59	kilometres	99.91	how	98.93
whose	92.14	can	97.08	3,500	93.82	unless	93.46	females	94.98	mountains	98.83
russian	90.91	ter	95.37	18%	93.72	71	92.72	Cass	93.62	where	98.68
cardinal	90.25	latitude	93.81	20%	93.45	nor	92.46	Kazakhstan	85.19	hundred	97.44
orient	89.83	dalai	92.04	whose	91.79	where	91.30	trillion	85.14	ship	96.52
german	89.03	samurai	90.52	11%	88.79	besides	90.62	Dull	83.16	least	95.29
korean	87.77	granting	90.24	54%	88.34	correlation	88.56	Estonia	80.62	USA	92.90
fide	84.10	reich	90.10	supplemented	86.34	holds	87.90	wherein	79.12	who	92.67
atop	82.91	banning	88.47	50%	85.46	waived	87.12	sued	78.91	haven	91.65

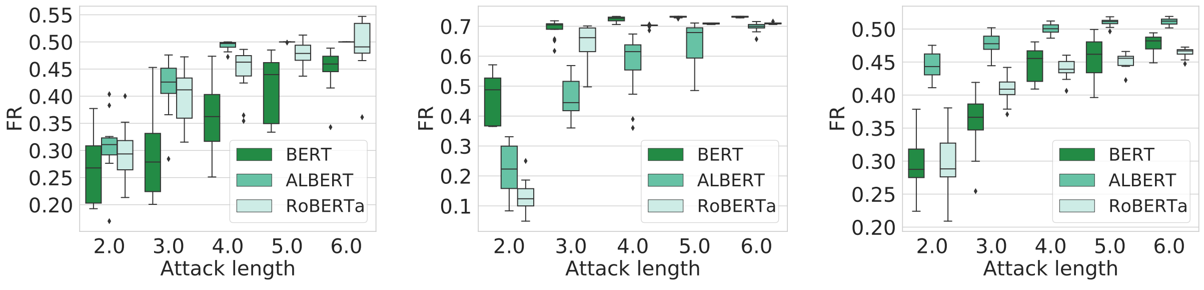
Table 3: PMI for the joint training and validation set of QNLI, E – Entailment, N – Not entailment

From/To	MRPC, W/S			QNLI, W/S			SST-2, W/S		
	ALBERT	BERT	RoBERTa	ALBERT	BERT	RoBERTa	ALBERT	BERT	RoBERTa
ALBERT	-	22.8/**53.5**	33.9/**69.1**	-	**45.6**/36.1	**39.0**/38.1	-	30.6/**37.5**	24.2/**34.7**
BERT	6.7/**17.2**	-	60.5/**67.8**	34.6/**36.3**	-	22.0/**36.0**	**31.0**/29.3	-	17.7/**24.1**
RoBERTa	15.3/**32.1**	55.5/**64.6**	-	20.4/**40.4**	18.0/**35.9**	-	**34.1**/26.2	36.9/**37.7**	-

Table 4: Transferability comparison between LUATs (S) and UATs of Wallace et al. (2019) (W). We report FR after performing the attack transfer between different models trained on a fixed dataset for the trigger length $L = 3$.

From/To	ALBERT, W/S			BERT, W/S			RoBERTa, W/S		
	MRPC	QNLI	SST-2	MRPC	QNLI	SST-2	MRPC	QNLI	SST-2
MRPC	-	9.5/**21.9**	5.4/**32.5**	-	**18.3**/17.6	**11.8**/9.4	-	**31.4**/25.6	5.3/**6.0**
QNLI	13.6/14.0	-	9.0/**31.7**	32.9/**35.5**	-	2.8/**4.7**	22.7/**40.9**	-	2.9/**8.9**
SST-2	8.6/**22.8**	31.3/**36.0**	-	29.7/**64.8**	**27.5**/21.2	-	**18.1**/13.9	14.0/**15.9**	-

Table 5: Transferability comparison between LUATs (S) and UATs of Wallace et al. (2019) (W). We report FR after performing the attack transfer between different datasets for a fixed model and the trigger length $L = 3$.



(a) SST.

(b) MRPC.

(c) QNLI.

Figure 3: FR for optimal q depending on the trigger length for the most vulnerable sets. The plots show FR saturation which is achieved for short enough triggers.

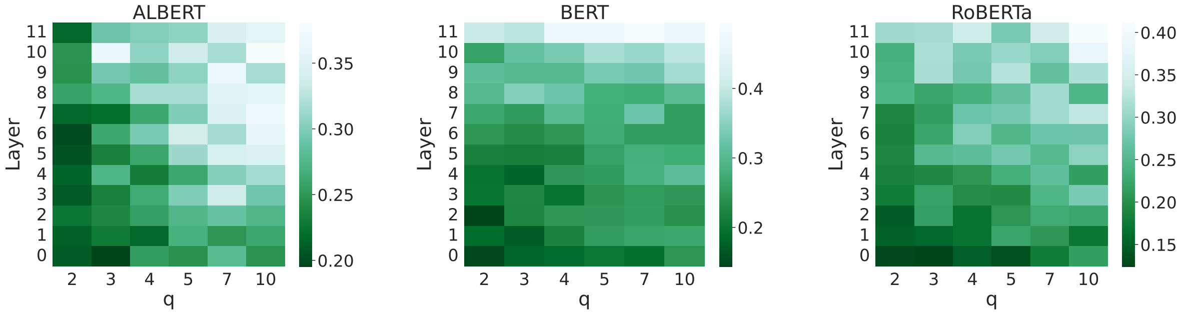


Figure 4: FR dependence on LUATs hyperperameters. The efficiency of our attack increases with q and the layer number.

is done for simplicity. In this setting, we suppose that an attacker has access to the data and also to the input and output of an attacked model. When transferring the UATs, the best trigger obtained with a source model is taken and then applied directly to a target model on a test set. For LUATs, however, one can come up with a better way. Since a target model is different from a source one, it is not necessary that the same values of q and l would provide the best pick for the transfer. Therefore, since an attacker has access to the forward pass of a target model, we evaluate with it the fooling rate of the best triggers found for each value of q and l on a validation set and then apply the best option to a test set to get the final score. For simplicity, we fix $q = 10$, which according to the Fig. 4, is the best alternative on average. The final results are presented in the Tab. 4. One can see that in most cases, we outperform the UATs with an average gain of 9.3%.

Transferability: task-to-task. The fact that the universal triggers can generalize to other models trained on the same task seems natural since, in this case, it is highly probable that a source and a target model would acquire the same biases. A more complicated situation is when one tries to transfer triggers between different tasks. In this setting, an attacker looks for more fundamental task-independent flaws of a model, which can be explained, e.g., by a bias appearing during the pretraining phase. In order to examine this capacity of the triggers, we perform their transferring between different datasets for each of the considered models. For measuring the performance of the UATs, as in the previous case, the best trigger obtained on a source task is transferred. On the other hand, for the layerwise approach, one, under a reasonable assumption, can still suggest a natural way to choose the most appropriate for transfer trigger among the best triggers corresponding to different values of q and l. Namely, we suppose that although attackers do not have access to data on which a target model was trained, they know a task for which it was trained. It means that they know whether it is sentiment classification, paraphrase identification, etc. (Savchenko et al., 2020). If this is the case, they can generate data corresponding to the task of interest. In order to mimic such a situation, for each of the datasets (SST-2, MRPC, QNLI), we select an auxiliary dataset collected for the same task.

The map is the following: SST-2 - IMDb (Maas et al., 2011), MRPC - PAWS (Zhang et al., 2019), QNLI - WikiQA (Yang et al., 2015). We sample subsets of sizes 64, 128, and 256 from the auxiliary datasets and consider them as the data generated by an attacker. Again, fixing $q = 10$ for simplicity, we evaluate the best trigger obtained for each layer on a corresponding auxiliary subset, performing inference on a target model. Herewith, each subset is sampled five times, and we report the average score for the size of 256, which appears to be the best option. The final results for both approaches are shown in the Tab. 5. Our approach demonstrates the average improvement of 7.1% in fooling rate over the vanilla UATs. The reason for that might be related to the fact that if this kind of triggers (task-independent) is indeed related to the pretraining phase, in order to find them, one instead should perturb the lower or middle layers since the higher layers' weights can change a lot after fine-tuning. The LUATs which appear to be most successful at transferring are presented in the Tab. 8 and 9 of the Appendix B for model-to-model and task-to-task transfers correspondingly.

4 Related Work

There are already quite a few works devoted to the universal attack on NLP models in the literature. We briefly discuss them here.

Ribeiro et al. (2018) proposed a sample-agnostic approach to generate adversarial examples in the case of NLP models by applying a semantics-preserving set of rules consisting of specific word substitutions. However, found rules were not completely universal, resulting in model prediction change only for $1\% - 4\%$ out of targeted samples.

Similarly to Wallace et al. (2019), the realization of text adversarial attacks as short insertions to the input were proposed by Behjati et al. (2019). Though, they did not perform the search over the whole vocabulary for each word position in the trigger but instead exploited cosine-similarity projected gradient descent, which does not appear that efficient in the sense of attack performance.

Adversarial triggers generated by the method proposed by Wallace et al. (2019) in general turned out to be semantically meaningless, which makes them easier to detect by defence systems. An attempt to make triggers more natural was undertaken by Song et al. (2021). Leveraging an adversarially regularized autoencoder (Zhao et al., 2018) to generate the triggers, they managed to improve their semantic meaning without significantly decreasing the attack efficiency.

Another interesting direction is to minimize the amount of data needed for finding UAPs. Singla et al. (2022) created representatives of each class by minimizing the loss function linear approximation over the text sequences of a certain size. Afterwards, adversarial triggers were appended to these class representatives and the rest of the procedure followed Wallace et al. (2019). Although no data was used explicitly for training the attack, this approach demonstrated solid performance on considered datasets.

5 Conclusion

We present a new layerwise framework for the construction of universal adversarial attacks on NLP models. Following Wallace et al. (2019), we look for them in the form of triggers; however, in order to find the best attack, we do not perturb the loss function but neural network intermediate layers. We show that our approach demonstrates better performance when the triggers are transferred to different models and tasks. The latter might be related to the fact that in order to be transferred successfully between different datasets, a trigger should reflect network flaws that are task-independent. In this case, reducing the attack search to perturbation of the lower or middle layers might be more beneficial since the higher layers are highly influenced by fine-tuning. We hope this method will serve as a good tool for investigating the shortcomings of language models and improving our understanding of neural networks' intrinsic mechanisms.

We would like to conclude by discussing the potential risks. As any type of technology, machine learning methods can be used for good and evil. In particular, adversarial attacks can be used for misleading released machine learning models. Nevertheless, we think that revealing the weaknesses of modern neural networks is very important for making them more secure in the future and also for being able to make conscious decisions when deploying them.

6 Limitations and future work

Our approach to universal text perturbations suffers from linguistic inconsistency, which makes them easier to detect. Therefore, as the next step of our research, it would be interesting to investigate the possibility of improving the naturalness of adversarial triggers without degradation of the attack performance in terms of the fooling rate.

While the proposed approach outperforms the UATs of Wallace et al. (2019) in the transferability task, we should highlight that the additional hyperparameters adjustment plays a crucial role, and one could suggest validation procedure refinement for a more fair comparison. Also, for both direct and transferability settings, a more comprehensive range of models should be examined, including recurrent (Yuan et al., 2021) and transformer architectures, e.g., T5 (Raffel et al., 2020), XLNet (Yang et al., 2019), GPT family models (Radford et al., 2019; Brown et al., 2020).

Another direction of improvement is related to the fact that sometimes the found triggers can change the ground truth label of samples they are concatenated to if, e.g., they contain words contradicting the true sense of a sentence. It would be interesting to analyze how often this happens and develop an approach to tackle this issue.

Finally, it would be interesting to investigate the dependence of attack efficiency on the size of a training set and compare it with the so-called data-free approaches, such as the one proposed by Singla et al. (2022).

Acknowledgements

The work was supported by Ministry of Science and Higher Education grant No. 075-10-2021-068.

References

Melika Behjati, Seyed-Mohsen Moosavi-Dezfooli, Mahdieh Soleymani Baghshah, and Pascal Frossard. 2019. Universal adversarial attacks on text classifiers. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7345–7349. IEEE.

David W Boyd. 1974. The power method for l^p norms. *Linear Algebra and its Applications*, 9:95–101.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems (NeurIPS)*, 33:1877–1901.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. 2021. Towards robustness against natural language word substitutions. *arXiv preprint arXiv:2107.13541*.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *Proceedings of Security and Privacy Workshops (SPW)*, pages 50–56. IEEE.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5747–5757, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Valentin Khrulkov and Ivan Oseledets. 2018. Art of singular vectors and universal adversarial perturbations. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8562–8570. IEEE.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*.

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*.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL): Human language technologies*, pages 142–150.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. Association for Computational Linguistics.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pages 119–126.

Vladislav Mosin, Igor Samenko, Borislav Kozlovskii, Alexey Tikhonov, and Ivan P Yamshchikov. 2023. Fine-tuning transformers: Vocabulary transfer. *Artificial Intelligence*, page 103860.

Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. 2016. Crafting adversarial input sequences for recurrent neural networks. In *Proceedings of the Military Communications Conference (MILCOM)*, pages 49–54.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics.

Andrey Savchenko, Anton Alekseev, Sejeong Kwon, Elena Tutubalina, Evgeny Myasnikov, and Sergey Nikolenko. 2020. Ad lingua: Text classification improves symbolism prediction in image advertisements. In *Proceedings of the 28th International Conferences on Computational Linguistics (COLING)*, pages 1886–1892.

Yaman Kumar Singla, Swapnil Parekh, Somesh Singh, Changyou Chen, Balaji Krishnamurthy, and Rajiv Ratn Shah. 2022. MINIMAL: Mining models for universal adversarial triggers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11330–11339.

Liwei Song, Xinwei Yu, Hsuan-Tung Peng, and Karthik Narasimhan. 2021. Universal adversarial attacks with natural triggers for text classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (ACL): Human Language Technologies*, pages 3724–3733. ACL.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. ACL.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Proceedings of Thirty-fifth Conference on Neural Information Processing Systems (NeuriPS) Datasets and Benchmarks Track (Round 2)*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. 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. *Advances in neural information processing systems (NeurIPS)*, 32.

Lifan Yuan, Yichi Zhang, Yangyi Chen, and Wei Wei. 2021. Bridge the gap between CV and NLP! A gradient-based textual adversarial attack framework. *arXiv preprint arXiv:2110.15317*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Junbo Zhao, Yoon Kim, Kelly Zhang, Alexander Rush, and Yann LeCun. 2018. Adversarially regularized autoencoders. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 5902–5911. PMLR.

A Linear problem solution on $\mathcal{S}$

Let us consider the following linear problem with a cost matrix C:

$$\max_{W \in \mathcal{S}} \langle C, W \rangle.$$

We need to construct the Lagrangian L and the consequent dual problem to obtain its solution.

$$\begin{aligned} L &= -\operatorname{Tr}((C+M)^\top W) + \lambda^\top (W\mathbb{1} - \mathbb{1}) \\ &= -\operatorname{Tr}((C+M-\lambda\mathbb{1}^\top)^\top W) - \lambda^\top \mathbb{1} \end{aligned}$$

where λ and M are the Lagrange multipliers. From KKT conditions, we have:

$$\begin{gathered} C + M - \lambda\mathbb{1}^\top = O, \\ M \geq 0, \quad M \cdot W = 0, \quad W\mathbb{1} = \mathbb{1}, \end{gathered}$$

where $M \cdot W$ means elementwise multiplication. As a result, we obtain the following dual problem:

$$\begin{aligned} &\max -\lambda^\top \mathbb{1}, \\ &s.t. \ -\lambda\mathbb{1}^\top \leq -C. \end{aligned}$$

Under the assumption that each row of the cost matrix C has a unique maximum, the closed-form solution will take the form:

$$\lambda_i = \max_j C_{ij}, \forall i,$$

and the corresponding primal solution

$$W = \begin{cases} W_{ij} = 1, \ j = \operatorname{argmax}_j C_{ij} \forall i, \\ W_{ij} = 0, \ \text{otherwise}. \end{cases}$$

Otherwise, if any row i of C violates the above assumption by having $k > 1$ maximal elements with indices $\{j_1, \dots, j_k\}$, then

$$W_{ij} = \frac{1}{k}, \quad \forall k \in \{j_1, \dots, j_k\}.$$

B Tables and plots

In this appendix, we present the results of

- the ablation study on the top-k and beam search parameters (see Fig. 5),
- the results of trigger analysis with PMI (see Tab. 6 and 7) for SST-2 and MRPC datasets,
- the LUATs which appear to be the best for model-to-model and task-to-task transfers (see Tab. 4 and 5),
- the cases when shorter triggers appear to be better than longer ones (see Tab. 10),
- the examples of the top-20 obtained triggers.

Concerning the transfer triggers, one can see that sometimes the same triggers efficiently break different models trained on different datasets, e.g., 'WHY voted beyond' (FR = 40.4 for ALBERT trained on QNLI, FR = 45.9 for BERT trained on QNLI, FR = 40.9 for RoBERTa trained on MRPC), 'unsuitable improper whether' (FR = 37.5 for BERT trained on SST-2, FR = 34.7 on RoBERTa trained on SST-2, FR = 22.8 for ALBERT trained on MRPC). This can serve as evidence of the high generalizability of universal adversarial triggers.

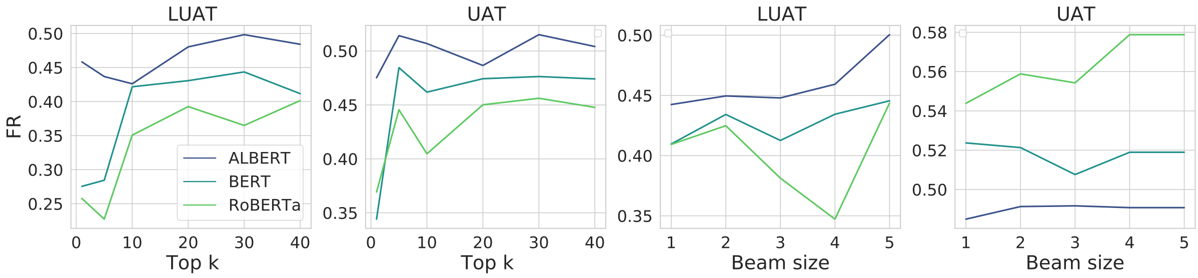


Figure 5: The results of the ablation study with respect to the top-k and the beam size parameters on QNLI dataset.

	BERT				ALBERT				RoBERTa		
N	Rank	P	Rank	N	Rank	P	Rank	N	Rank	P	Rank
worst	99.96	also	99.13	fails	99.70	coming	94.40	neither	99.44	powerful	99.95
stupid	99.87	drama	98.51	devoid	99.39	foremost	82.79	failure	99.23	enjoyable	99.94
fails	99.66	definitely	96.00	failure	98.90	behalf	81.14	whether	97.22	beautiful	99.91
poor	99.48	shows	95.81	whether	97.22	warmed	28.42	despite	95.17	remarkable	99.78
neither	99.35	walk	94.89	unless	94.21	placement	10.03	considering	89.70	refreshing	99.76
badly	99.32	ranks	93.12	irrelevant	93.76	irrelevant	6.30	because	87.80	thriller	99.62
crap	98.91	but	86.00	placement	90.06	unless	5.91	never	80.57	warmth	99.55
pacing	97.94	these	84.96	warmed	71.65	whether	2.78	despite	80.48	impressive	99.46
whether	97.03	wherein	78.49	behalf	19.60	failure	1.11	ball	74.50	creative	99.34
every	96.81	weeks	59.58	foremost	17.79	devoid	0.61	notice	70.93	delightful	99.26

Table 6: PMI for the joint training and validation set of SST-2, N – Negative, P – Positive

	BERT				ALBERT				RoBERTa		
N	Rank	E	Rank	N	Rank	E	Rank	N	Rank	E	Rank
succeeded	97.09	never	97.74	ashamed	95.06	after	99.48	waves	95.42	fire	98.91
seeing	95.94	least	96.50	stumbling	87.91	under	99.45	aside	95.42	against	98.65
merits	88.23	killing	96.19	tire	87.91	because	98.89	harming	88.06	water	95.89
longest	88.23	prison	95.58	cutting	77.42	since	98.27	skepticism	88.06	health	95.86
feeling	78.82	much	94.49	declined	76.26	how	97.70	burns	88.06	pleaded	95.19
raped	77.95	welcome	94.04	boise	70.31	whether	97.51	survives	88.06	nothing	93.25
batting	77.95	behind	93.65	fury	66.42	should	97.12	votes	76.46	won	92.13
jail	74.66	born	92.60	slay	55.80	cost	96.57	justice	70.62	NATO	91.82
banning	70.64	needed	92.26	reactions	52.67	serious	96.38	ressing	65.60	suffering	88.81
backing	70.64	nothing	91.30	disapprove	52.67	guilty	96.31	deceived	48.45	happens	87.63

Table 7: PMI for the joint training and validation set of MRPC, N – not equivalent, E – equivalent

	MRPC		QNLI		SST-2	
	Trigger	l	Trigger	l	Trigger	l
A → B	they ended and	10	preis much $100, 000	3	unsuitable improper whether	10
A → R	rostov she blushed	1	$100, 000 what simulate	2	unsuitable improper whether	10
B → A	nothing pains kilograms	1	whichever thirds lithuanian	9	folding worse as	5
B → R	suffered sins declined	11	ibly semester longest	8	folding worse as	5
R → A	Avoid water taps	10	WHY voted beyond	2	decayingjuryNeither	1
R → B	Avoid water taps	10	WHY voted beyond	2	surprisingly refreshing lest	11

Table 8: The best-performed trigger-layer pairs for model-to-model transferability results presented in Tab. 4 for ALBERT (A), BERT (B), RoBERTa (R), where l – the perturbed layer.

	ALBERT		BERT		RoBERTa	
	Trigger	l	Trigger	l	Trigger	l
M → Q	nobody reminds austro	8	is blood corrupt	9	history doubtless beyond	8
M → S	failing forcing the	4	ching stiff punishments	8	history doubtless beyond	8
Q → M	or widen further	10	whichever thirds lithuanian	9	WHY voted beyond	2
Q → S	trillion unless marylebone	5	ant trees romanized	5	wherein perished supra	7
S → M	unsuitable improper whether	10	seemed wiped whoever	9	Eating welcome respecting	7
S → Q	renumbered littered neither	8	ogarily diminished	4	Crystal tasty ain	9

Table 9: The best-performed trigger-layer pairs for task-to-task transferability results presented in Tab. 5 for MRPC (M), QNLI (Q), SST-2 (S), where l – the perturbed layer.

Model	L	**FR Val.**	**FR Test**	**Trigger**
UAT, BERT	2	38.6	38.4	neither nor
UAT, BERT	4	35.4	35.0	situation nonetheless resulted nor
Ours, ALBERT, $q = 7$, $i = 4$	5	24.8	24.8	regretted joyahbwv although doubted
Ours, ALBERT, $q = 4$, $i = 8$	6	21.4	21.4	tremendous despair towedtrue 1985, doubted

Table 10: The examples of cases when a shorter-length trigger is more successful than a longer one. In both cases, triggers were fitted on MNLI.

Dataset	**Model**	**Triggers**
MNLI	ALBERT	hamas doubted; neither motioned; i doubted pronoun; cursing neutron unless; transportation workers unless preferring;
	BERT	but neither; remarkably neither; of neither nor; get neither outta even; backdrop but and neither; ft still maintained not whether
	RoBERTa	slideshow Neither; HELL Neither; THERE HAS NEVER; ohan Never nor; Fifa VERY NEVER Whether
MRPC	ALBERT	unless if; whereas and; they ended and; unless stumbling out and; commerce corroborat declined and; and siblings live because
	BERT	against attempt; prepares worse conflicts; was priesthood killing contrary; each mortally duel harta; ring opposed her homosexuality
	RoBERTa	Drop until; actresses won awards; Enhanced Population Died Low; Harlem Recommended Submit BLM Question; hum DOWN burns firing Bloody Hyp
QNLI	ALBERT	and 40%; averaged percentage; 88% finance 11%; vittorio whom relinquished; cristo, sharing whom reissued; downloadable bilingual why cancellation
	BERT	nine charities; carlos orient whom; whose reich declaring; gor german which emperor; maya commuted whose ballet
	RoBERTa	nineteen countries; Nearly trillions trillion; Thousand hundred trillion; how MUCH mountains Cass; gui why sued awaits Reviews;
SST-2	ALBERT	flight unless; suck unnecessary; skidded irrelevant whether; failure placement unless; just whine worthless; fails subdistrict picture
	BERT	degraded whether; dissatisfied neither; television failed whether; definitely worst wherein whoever; crap stupid feed whereby whether
	RoBERTa	powerfully refreshing; Beautiful enjoyable; surprisingly refreshing lest; Crystal importantly beautiful considering; thriller cool because whereas

Table 11: WARNING: THE CONTENT OF THIS TABLE MIGHT BE OFFENSIVE, AND IT DOES NOT REFLECT THE AUTHORS' OPINION. The examples of LUATs depending on a dataset and a model. Triggers were selected manually from top-20 per length.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
6

☑ A2. Did you discuss any potential risks of your work?
5

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☑ A4. Have you used AI writing assistants when working on this paper?
Grammarly for check spelling

B ☑ Did you use or create scientific artifacts?

3

☑ B1. Did you cite the creators of artifacts you used?
3

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We used freely available models and dataset

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
We did not discuss the indented use of the models and datasets which we used, since these are the models and datasets are known to everyone and their use is free.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We did not collect any data, and we used only publicly available datasets.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
We did not collect any data. The documentation for the datasets and models which we are used are publicly available.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
3

C ☑ Did you run computational experiments?

3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
3, Appendix B

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
3, Appendix B

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
3

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.