

IMPROVED AND EFFICIENT TEXT ADVERSARIAL ATTACKS USING TARGET INFORMATION

Mahmoud Hossam, Trung Le, He Zhao, Viet Huynh & Dinh Phung

Department of Data Science and AI, Faculty of Information Technology

Monash University

Clayton 3800, Victoria, Australia

{mhossam, trunglm, ethan.zhao, viet.huynh, dinh.phung}@monash.edu

ABSTRACT

There has been recently a growing interest in studying adversarial examples on natural language models in the black-box setting. These methods attack natural language classifiers by perturbing certain important words until the classifier label is changed. In order to find these important words, these methods rank all words by importance by querying the target model word by word for each input sentence, resulting in high query inefficiency. A new interesting approach was introduced that addresses this problem through interpretable learning to learn the word ranking instead of previous expensive search. The main advantage of using this approach is that it achieves comparable attack rates to the state-of-the-art methods, yet faster and with fewer queries, where fewer queries are desirable to avoid suspicion towards the attacking agent. Nonetheless, this approach sacrificed the useful information that could be leveraged from the target classifier for the sake of query efficiency. In this paper we study the effect of leveraging the target model outputs and data on both attack rates and average number of queries, and we show that both can be improved, with a limited overhead of additional queries.

1 INTRODUCTION

Most of the existing black-box attack methods on natural language models directly query the target model to rank sentence words for replacement. This procedure is expensive in terms of number of queries needed to the target model, and the number of queries increases with increasing input length. In addition, the increased number of queries to the target classifier is not desirable in black-box settings, where it can raise the suspicion towards the attacking agent. Recently, several methods were introduced to improve the query efficiency, including (Garg & Ramakrishnan, 2020; Li et al., 2020) and *Explain2Attack* (Hossam et al., 2020), where an interpretable model was trained to learn word importance ranking for synonym replacement.

Unlike the other methods, *Explain2Attack* addresses the word ranking query inefficiency, which is closely related to the target model, and causes a hurdle into efficiently crafting the adversarial example. In *Explain2Attack* model, a *selector* network \mathcal{E} is trained to learn the important words by approximating the performance of a *substitute classifier* F_b . Under this architecture, *Explain2Attack* was able to achieve comparable attack rates to the state-of-the-art method (Jin et al., 2019), while reducing the number of queries.

However, the word ranking step in *Explain2Attack* was trained on the substitute dataset and classifier, which are both different from the target classifier and the target dataset. The substitute training was designed in this way in order to limit the access to the target model through extensive queries, and to its training dataset D_t^{train} . The key intuition is that as long as the chosen dataset for the substitute training dataset D_b comes from a similar domain to the target training dataset, then the learned word ranking model should perform well enough on test sentences from the target dataset domain.

Although this setting leads to competitive attack rates and reduced queries, it is still a restriction on the substitute training procedure. Additionally, in some scenarios, the access to target model data

might be available. Thus, by relaxing restrictions on the access to possible target model information (both outputs and data domain), we might be able to achieve better attack rates with a marginal overhead of additional queries used for the substitute training.

In this paper, we propose two modifications to this interpretable approach to better train the selector network while benefiting from available target information. Specifically, we do not use the substitute dataset labels used for selector training as in (Hossam et al., 2020). Instead, we replace them with output predictions from the target classifier under two settings: the target model is given the substitute dataset sentences $\mathcal{X}_b \in D_b$, or the target model is given the target test set $\mathcal{X}_t^{\text{test}} \in D_t^{\text{train}}$.

The advantage are twofold: *i) we relax the requirement of finding a suitable substitute data domain to learn the word ranking*, and *ii) we achieve both improved attack rates and reduced average number of queries, with limited effect on the number of overhead queries*. We describe in details the proposed modifications and their intuitions, and provide experimental results for different classification models and datasets.

2 BACKGROUND

Here we revise the original setting of Explain2Attack (Hossam et al., 2020): let a target model F_{target} be trained on some target training dataset $D_t^{\text{train}} = \{\mathcal{X}_t^{\text{train}}, \mathcal{Y}_t^{\text{train}}\}$ and testing set $D_t^{\text{test}} = \{\mathcal{X}_t^{\text{test}}, \mathcal{Y}_t^{\text{test}}\}$. A substitute model, SUB, is then trained to learn the word importance scores. SUB is trained using a dataset that is close enough to the target model dataset called the substitute dataset, $D_b = \{\mathcal{X}_b, \mathcal{Y}_b\}$. The substitute model itself contains two sub-networks called the *substitute classifier* F_b and the *selector* network \mathcal{E}_θ . After training, performing inference on the selector network $\mathcal{E}_\theta(X)$ using input sentences yields the desired word importance scores for these sentences. During the substitute training procedure, the substitute classifier F_b is trained to correctly predict the label $Y \in \mathcal{Y}_b$ from an input sentence $X_s \sim \mathcal{E}_\theta(X)$, where $X \in \mathcal{X}_b$, and X_s is the most important selected k words for some input X . After substitute training is finished, $F_b(\cdot)$ can be discarded, since we are only interested in the selector $\mathcal{E}(X)$.

3 METHOD

We change the original setting of Explain2Attack to incorporate the target classifier predictions outputs in the selector network training. Specifically, we consider two black-box settings: first, the substitute networks are trained without access to the target test set, and second, they are trained with access to the target test set.

In details, we want to replace the labels \mathcal{Y}_b that come from the substitute dataset D_b with output labels or probabilities from the target classifier F_{target} . This way, the training of F_b , and most importantly, the selector \mathcal{E}_θ , will be trained such that it learns or imitates the behaviour of the target classifier. This requires querying the target model $F_{\text{target}}(\tilde{X})$ with some input \tilde{X} to return target classifier predictions. The choice of the inputs to the target model \tilde{X} allows to different settings: *i)* We either choose \tilde{X} to come from the substitute dataset \mathcal{X}_b , or *ii)* from the target test-set sentences $\mathcal{X}_t^{\text{test}}$. Below we describe both settings in details and the possible use cases for both.

3.1 SUBSTITUTE TRAINING WITHOUT ACCESS TO THE TARGET TEST SET

In this setting, we use the target model’s predictions to come from substitute sentences input $X_b \in \mathcal{X}_b$ instead of using the substitute dataset labels \mathcal{Y}_b . We use the prediction probabilities from F_{target} for the substitute training, where the loss function for training both F_b and \mathcal{E}_θ becomes:

$$\mathcal{L}_{\text{subs.domain}}(F_b, Y_{\text{target}}) = \mathbb{E}_{X_b \in D_b} \left[\|F_{b,\phi}(\mathcal{E}_\theta(X_b)) - F_{\text{target}}(X_b)\|_2^2 \right], \quad (1)$$

where D_b is the substitute dataset, $F_b(\mathcal{E}_\theta(\cdot))$ is the substitute classifier output, and Y_{target} is the output of the target classifier F_{target} , both containing the probabilities for all available classes. This loss is the mean square error (MSE) between the substitute classifier and the target classifier outputs. Unlike other losses like the KL-divergence, the information about the other classes in the soft label

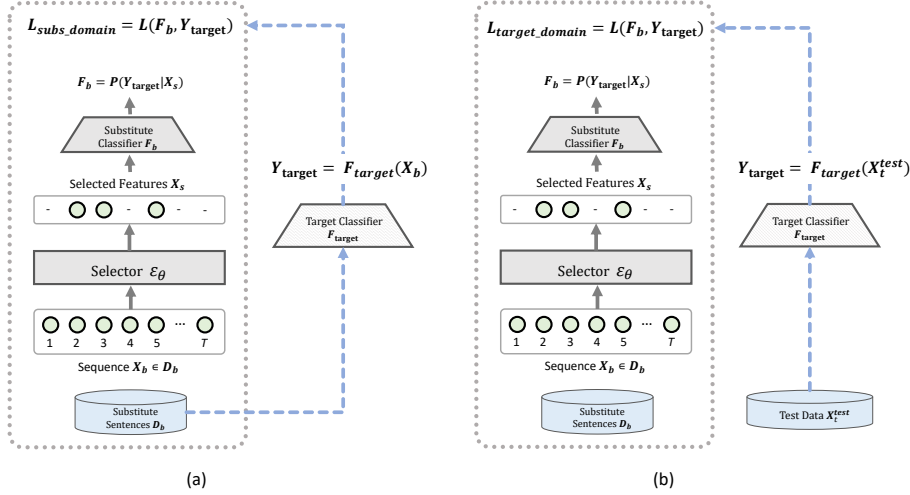


Figure 1: Overview of modified Explain2Attack training with target model information via a) the substitute sentences, or b) the target test set.

outputs (all classes probabilities) are included. Therefore the purpose of using the MSE loss is to train the substitute classifier such that its real-valued output predictions (all of the classes probabilities) are optimized to resemble the target classifier prediction outputs. Therefore, the selector network is trained to help the substitute classifier resemble the target classifier behavior. Fig. (1a) shows the substitute training procedure under this setting. This setting is suitable when the target test set is not known for the attacker, which is the common scenario of black-box attacks on pretrained models.

3.2 SUBSTITUTE TRAINING WITH ACCESS TO THE TARGET TEST SET

In this setting, we use the target model’s outputs to come from a part of the target test set $X_t^{\text{test}} \in \mathcal{X}_t^{\text{test}}$. The rationale behind this setting is that we consider the cases when the attacker happens to have access to the target dataset, or that the user of the target model gets to choose the training dataset, yet does not have access to the final target model parameters. Similar to the previous setting, we use the probability outputs from F_{target} for the substitute training, where the loss function becomes:

$$\mathcal{L}_{\text{target_domain}}(F_b, Y_{\text{target}}) = \mathbb{E}_{X_b \in D_b, X_t^{\text{test}} \in \mathcal{X}_t^{\text{test}}} \left[\|F_{b,\phi}(\mathcal{E}_\theta(X_b)) - F_{\text{target}}(X_t^{\text{test}})\|_2^2 \right]. \quad (2)$$

It is important to mention that in Eq.(2) X_t^{test} could be used instead of X_b in principle, however, we show that even using data other than the target test still improves the results. Fig. (1b) shows the substitute training procedure under this setting.

4 EXPERIMENTS AND DISCUSSION

We employ target classifier predictions under these two settings on sentiment classification task using WordCNN and WordLSTM target classifiers. For all of the experiments, we used the same datasets, substitute datasets, and target classifier parameters and hyperparameters that were used in (Hossam et al., 2020).

To evaluate the performance of the two proposed methods described above, we report the adversarial accuracy (Adv_Acc), the average number of queries (Adv_Queries), and compare with the original Explain2Attack reported in (Hossam et al., 2020). In detail, in tables 1 and 2 we report the results for incorporating the target model predictions for training Explain2Attack selector without access to the target test set and with access to it, respectively. For both Adv_Acc and Adv_Queries, the lower number is the better, indicated by the \downarrow symbol.

For the first method, we use the exact setup as in (Hossam et al., 2020) for the experiments and the datasets, except that the substitute labels \mathcal{Y}_b were replaced with the target classifier predictions from the substitute sentences, and Eq. (1) was used for training the selector. For the second method, \mathcal{Y}_b were replaced with the target classifier predictions from the target test set, and Eq. (2) was used to train the selector. For the set of experiments in Table 2, only a small portion of the test set was used during the selector training, where 5000 out of 25,000 sentences were used.

For the set of experiments in Table 1, the first method was used to attack the target classifier. We can see that in three out of four experiments, that the attack rate was improved compared to the original baseline. This demonstrates the added benefit of incorporating the target model predictions in the training process. Moreover, we find that the average number of queries needed was also improved in these three cases.

Table 1: Performance metrics for Explain2Attack (E2A) with the selector trained on target model predictions given substitute dataset sentences.

Classifier		WordCNN		WordLSTM	
Target Model		IMDB	Amazon MR	IMDB	Amazon MR
Clean_Acc.		87.32	90.16	88.78	91.44
<i>(Substitute Data)</i>		<i>(Yelp)</i>	<i>(IMDB)</i>	<i>(Amazon MR)</i>	<i>(IMDB)</i>
Adv_Acc. ↓	E2A	0.59	4.12	0.05	2.51
	E2A- $F(X_b)$ (ours)	0.57	4.11	0.08	2.37
Avg_Queries ↓	E2A	402.5	351.7	440.2	368.7
	E2A- $F(X_b)$ (ours)	345.6	337.8	652.7	336.0

Table 2: Performance metrics for Explain2Attack (E2A) with the selector trained on target predictions given target test set sentences (5K out of 25K target test sentences used).

Classifier		WordCNN		WordLSTM	
Target Model		IMDB	Amazon MR	IMDB	Amazon MR
Clean_Acc.		87.32	90.16	88.78	91.44
Adv_Acc. ↓	E2A	0.59	4.12	0.05	2.51
	E2A- $F(X_t^{\text{test}})$ (ours)	0.56	4.05	0.05	2.34
Avg_Queries ↓	E2A	402.5	351.7	440.2	368.7
	E2A- $F(X_t^{\text{test}})$ (ours)	382.6	352.2	444.4	357.2

The improvement of the average number of queries is of particular interest in our setting, as it relates to the quality of word importance ranking learned during substitute training. Specifically, the baselines TextFooler (Jin et al., 2019) and the original Explain2Attack both perform the following procedure for generating an adversarial example: they perturb word by word, in order, from the most to least important words. In each of these perturbations, the target classifier is queried to check if its prediction label was changed. Therefore, the correctness of word ranking plays a key role in the total number of words that need to be perturbed, and consequently, the total number of queries. This means that the more important the selected words are to the target classifier, the fewer total words will be needed to be perturbed in order to change the final classification label. Thus, fewer queries will be needed. By looking on the results in tables 1 and 2, we can see that for most of the experiments, there is a consistent improvement in the number of queries. This observation suggests that the employing target model predictions encouraged the selector to learn more accurate word ranking according to its importance to the target classifier. Although there is some overhead of queries involved in training the selector in the first place, the later reduction in average queries needed per single attack might be of more value with the increased number of attacks.

In addition, we find that the overhead needed for substitute training queries can be reduced in the second method, where the target test set is used for output predictions from the target classifier. As we see in Table 2, the selector was trained only on 5000 test set sentences, out of the total 25,000 sentences in the test set. This also suggests that having access to the target test set might have an additional benefit on both attack rates and the number of queries. We included a comprehensive ablation study for different portion sizes used from the target dataset in Appendix A.2.

5 CONCLUSION

In this paper, we studied the effect of incorporating the target model domain data and outputs on attack rates and query efficiency in state-of-the-art black-box text attacks. By leveraging target model output predictions and test data, we presented two methods to improve the training process of the recent interpretable learning approach Explain2Attack. We use these target model outputs during the selector network training instead of the substitute labels used in the original setting. The intuition is to encourage the selector network to better learn the behaviour of the target classifier through its outputs, thus learning more accurate word importance ranking. The experiments show that on most of the selected datasets, there was an improvement in both attack rates and the average number of queries per attack. This study highlights the benefit of carefully incorporating both target model data and outputs in black-box attacks, while keeping a reduced average number of queries per adversarial example. In the future, we plan to further leverage the target classifier information by augmenting the substitute training set with the intermediate adversarial candidates and their target classifier outputs.

ACKNOWLEDGMENTS

This work was partially supported by the Australian Defence Science and Technology (DST) Group under the Next Generation Technology Fund (NTGF) scheme.

REFERENCES

- Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2890–2896, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1316. URL <https://www.aclweb.org/anthology/D18-1316>. A.1
- Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 50–56. IEEE, 2018. A.1
- Siddhant Garg and Goutham Ramakrishnan. Bae: Bert-based adversarial examples for text classification, 2020. 1, A.1
- Mahmoud Hossam, Trung Le, He Zhao, and Dinh Phung. Explain2Attack: Text Adversarial Attacks via Cross-Domain Interpretability. In *25th International Conference on Pattern Recognition, (ICPR)*, 2020. URL <https://arxiv.org/abs/2010.06812>. 1, 2, 4, A.1
- Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is BERT really robust? natural language attack on text classification and entailment. *CoRR*, abs/1907.11932, 2019. URL <http://arxiv.org/abs/1907.11932>. 1, 4, A.1, A.3
- Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. Adversarial examples for natural language classification problems, 2018. A.1
- Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. Bert-attack: Adversarial attack against bert using bert, 2020. 1, A.1
- Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011. A.3
- Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*, 2005. A.3

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1085–1097, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1103. A.1

Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I. Jordan. Greedy attack and gumbel attack: Generating adversarial examples for discrete data, 2019. URL <https://openreview.net/forum?id=ByghKiC5YX>. A.1

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pp. 649–657, 2015. A.3

A APPENDIX

A.1 RELATED WORK

Many methods were recently developed for black-box adversarial text attacks (Kuleshov et al., 2018; Yang et al., 2019; Gao et al., 2018; Ren et al., 2019; Jin et al., 2019; Garg & Ramakrishnan, 2020; Li et al., 2020). The main challenges for natural language adversarial attacks are the discrete nature of inputs, where defining meaningful perturbations is not straight forward, and the search space and complexity for finding the important words.

In details, several methods (Jin et al., 2019; Ren et al., 2019; Gao et al., 2018) have been developed that share similar general framework, where the attack starts by selecting the most important words/tokens to replace from a candidate sentence, followed by searching for some word replacement that can flip the classification label of the target model. However, some methods followed the heuristic optimisation approach, for example, (Alzantot et al., 2018) used a genetic algorithm to find the best sentence perturbation that fools the classifier.

Most of the aforementioned black-box methods use the word selection/replacement strategy. For instance, PWWS (Ren et al., 2019) proposes computing a word saliency score using output probabilities of the target model, while (Gao et al., 2018) computes sequential importance score based on forward and backward RNN probabilities at the current word position in the sentence. TextFooler (Jin et al., 2019) is a recent strong baseline for text attacks, where the method uses a modified procedure for word ranking that increases the ranking in label disagreement case. BERT-Attack and BAE-Attack (Li et al., 2020; Garg & Ramakrishnan, 2020) both improve on TextFooler synonym replacement by using a pretrained language model to generate suitable substitute words based on the surrounding context. This achieved higher attack rates and the number of queries is reduced.

The recent introduced approach Explain2Attack (Hossam et al., 2020) differs from previous work in solving the word ranking problem. Unlike other methods, instead of depending on the target model for word importance ranking, word importance scores are *learned*. The main differences of this approach compared existing ones are: i) word ranking complexity at inference time is of constant order in terms of input sentence length, therefore, it significantly reduces the number of queries needed to rank the words, as well as the running-time. ii) unlike existing methods, this approach is scalable with increased sentence lengths, since computing the scores is not dependent on word by word query of the target model. Moreover, this general approach can benefit from further query reduction in the synonym replacement phase by incorporating the pretrained language model technique in methods like BERT-Attack and BAE-Attack.

A.2 TARGET DATA SIZE ABLATION STUDY

We study the effect of the target set portion size used for the selector training, using 5000, 10,000, 15,000, and 20,000 samples out of the whole test set size of 25,000 samples. We perform the same set of experiments in Table 2 on these portions sizes and report the adversarial accuracies and average queries for all combinations of target models and datasets in Figures (2) and (3) compared to the original Explain2Attack.

For every target model and dataset combination, we can see that there is an improvement either in the adversarial accuracy (lower accuracy), or in the average number of queries (fewer queries), or

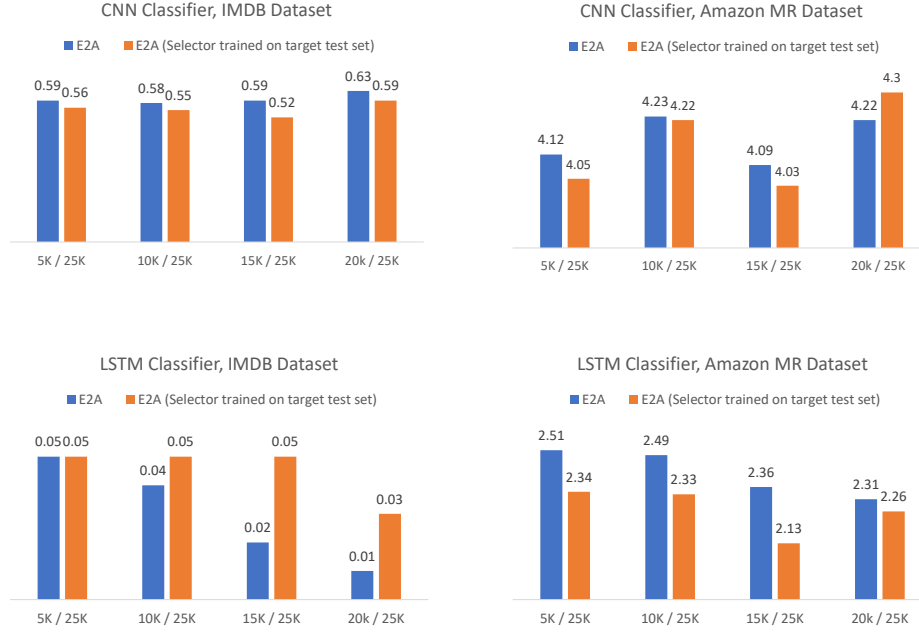


Figure 2: Adversarial Accuracies for Explain2Attack with the selector trained on a portion of the target test set. Each category represents the portion of the data used out of 25K samples. All combinations of target models and datasets are reported (*lower is better* ↓).

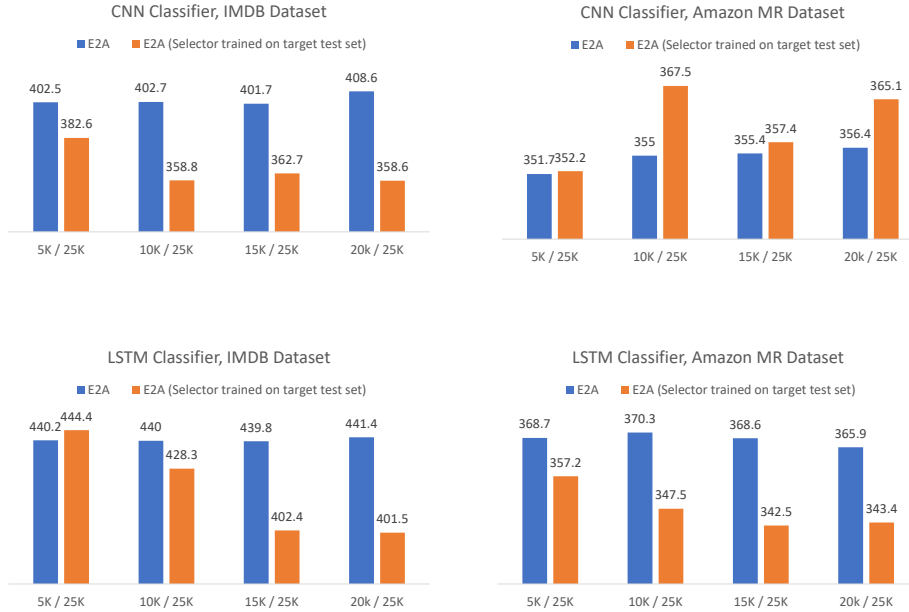


Figure 3: Average Queries for Explain2Attack with the selector trained on a portion of the target test set. Each category represents the portion of the data used out of 25K samples. All combinations of target models and datasets are reported (*lower is better* ↓).

both. Notably, we find that the best improvements happen mostly when the number of used samples is less than or equal to 15,000. This suggests either that there is a limit to the number of target test set samples that can be useful for the selector training, or that this behavior is just a special case on these selected datasets, and there might be a more general consistent behavior under other datasets with different sizes.

Similarly, we believe that the number of substitute dataset samples used in the first method for training the selector might have an impact on both attack rates and the number of queries. We look further to investigate the impact of both target and substitute dataset sizes on the overall performance by choosing more datasets with different sizes, yet we leave this comprehensive study for future work.

A.3 DATASETS

Here we briefly describe the datasets used in our experiments and their statistics:

- **IMDB:** Movie reviews for sentiment classification (Maas et al., 2011; Pang & Lee, 2005). The reviews have binary labels, either positive or negative.
- **Amazon MR:** Amazon polarity (binary) user reviews on movies, extracted from the larger Amazon reviews polarity dataset ¹.
- **Yelp Polarity Reviews:** Sentiment classification on positive and negative businesses reviews (Zhang et al., 2015). We mainly use this dataset as a substitute dataset when attacking other models.

In all of the datasets except Amazon MR, we follow the data preprocessing and partitioning in (Jin et al., 2019).

Table 3: Statistic of Used Datasets

Dataset	Train	Test	Avg. Length
IMDB	25K	25K	215
MR	9K	1K	20
Amazon MR	25K	25K	100
Yelp	560K	38K	152

¹<https://www.kaggle.com/bittlingmayer/amazonreviews>