Adv-OLM: Generating Textual Adversaries via OLM

Vijit Malik **Ashwani Bhat** **Ashutosh Modi**
Indian Institute of Technology Kanpur (IIT Kanpur)
{vijitvm, bashwani}@iitk.ac.in
ashutoshm@cse.iitk.ac.in

Abstract

Deep learning models are susceptible to adversarial examples that have imperceptible perturbations in the original input, resulting in adversarial attacks against these models. Analysis of these attacks on the state of the art transformers in NLP can help improve the robustness of these models against such adversarial inputs. In this paper, we present **Adv-OLM**, a black-box attack method that adapts the idea of **O**cclusion and **L**anguage **M**odels (OLM) to the current state of the art attack methods. OLM is used to rank words of a sentence, which are later substituted using word replacement strategies. We experimentally show that our approach outperforms other attack methods for several text classification tasks.

1 Introduction

In recent times, deep learning models have become pervasive across different domains. Many of the recent deep models have shown SOTA performance on a variety of NLP tasks (Wang et al., 2018). Consequently, deep models are being deployed in a variety of production systems for real-life applications. Hence, it becomes imperative to ensure the reliability and robustness of such models as it might pose a threat to security.

Recent studies have pointed out the vulnerability of deep models to adversarial attacks (Goodfellow et al., 2014). Adversarial attack comprises generating adversarial samples by performing small perturbations to the original input, making them imperceptible to humans while fooling the deep learning models to give incorrect predictions.

Adversarial attack on textual data is much more difficult due to the discrete nature of the text. The basic requirement of imperceptibility of perturbation by human judges is much more challenging in a language data setting. Therefore, the adversarial sample needs to be grammatically correct and semantically sound. Perturbations at word or character level that are perceptible to human judges have been explored in-depth (Ebrahimi et al., 2017; Belinkov and Bisk, 2017; Jia and Liang, 2017; Gao et al., 2018). Work on defense against misspellings based attacks (Pruthi et al., 2019) and use of optimization algorithms for attacks like genetic algorithm (Alzantot et al., 2018; Wang et al., 2019) and particle swarm optimization (Zang et al., 2020) have also been explored. With the rise of pre-trained language models, like BERT (Devlin et al., 2018) and other transformer-based models, generating human imperceptible adversarial examples has become more challenging. Wallace et al. (2019), Jin et al. (2019), and Pruthi et al. (2019) have explored these models from different perspectives.

Adversarial examples can be generated using black-box, where no knowledge about the model is accessible, and white-box, where information about the technical details of models are known. Generation of textual adversarial samples in a black-box setting consists of two steps 1) Finding words to replace in a sample (Word Ranking) 2) Replacing the chosen word (Word Replacement). Word Ranking is necessary to ensure that the word that contributes the most to the output prediction is considered as the candidate for replacement in the next step. Other constraints like generating semantically similar adversarial samples, human imperceptibility, and minimal perturbation percentage are also considered. Previous work has obtained word ranking by performing deletion of words (e.g., BAE-R (Garg and Ramakrishnan, 2020), TextFooler (Jin et al., 2019)), and replacement of words with *[UNK]* token (e.g., BERT-Attack (Li et al., 2020)) and then ranking the words based on the output logits difference.

Recently in the model explainability domain, the method of Occlusion and Language Models (OLM)

(Harbecke and Alt, 2020) has been proposed, the authors argue that the data likelihood of the samples obtained after either deleting the token or replacement with *[UNK]* token is very low, which makes these methods unsuitable for determining relevance of the word towards the output probability. The authors propose the use of language models for calculating the relevance of the words in a sentence. Taking inspiration from OLM, we propose **Adv-OLM**, a black box attack method, that adapts the idea of OLM (as the Work Ranking Strategy) to find the relevant words to replace. We empirically show that OLM provides a better set of ranked words compared to the existing word ranking strategies for the generation of adversarial examples.

We summarize our contributions as follows:

- We propose a new method Adv-OLM, to rank words for generating adversarial examples.
- We empirically show that Adv-OLM has a higher success rate and lower perturbation percentage than previous attacking methods.

The implementation for the proposed approach is made available at the GitHub repository: `https://github.com/vijit-m/Adv-OLM`.

2 Problem Formulation

We are given a corpus consisting of n input samples, $\mathbb{X} = \{x_1, \ldots, x_n\}$ with corresponding labels $\mathbb{Y} = \{y_1, \ldots, y_n\}$ and a trained classification model $f(f : \mathbb{X} \rightarrow \mathbb{Y})$ that maps an input samples to its correct label. We assume a black-box setting where the attacker can only query the classifier for output label probabilities for the given input. For an input sample $x \in \mathbb{X}$, the task is to construct an adversarial sample x' such that, $f(x) = y, \quad f(x') = y'$ with $\quad y \neq y'$, and $Similarity(x', x) \geq \epsilon$. Here, $Similarity : \mathbb{X} \times \mathbb{X} \rightarrow (0, 1)$ can be both the semantic and syntactic similarity function, and ϵ is the minimum similarity threshold. Ideally, the amount of perturbation should be minimized. The first step is to rank the words of the sample x. Based on the ranking, starting from the most important word, the word is replaced by some candidate word that keeps the perturbed sample x' semantically similar and grammatically sound but changes the output prediction.

3 Methodology

Adv-OLM uses the idea of Occlusion and Language Models to perform Word Ranking using both OLM and OLM-S methods. OLM uses a language model to sample some candidate instances for a word and then replaces the word. Let x_i be a word of the input x and $x_{\setminus i}$ be the incomplete input without this word. Then the OLM relevance score r given the prediction function f and label y is (Here f_y is the logit value corresponding to the label y.)

$$r_{f,y}(x_i) = f_y(x_i) - f_y(x_{\setminus i}) \tag{1}$$

Here, $f_y(x_{\setminus i})$ is not accurately defined and needs to be approximated since $x_{\setminus i}$ is the incomplete input. A language model p_{LM} generates input by predicting the masked word as $\hat{x_i}$ that is as natural as possible for the model and thus approximates to:

$$f_y(x_{\setminus i}) \approx \sum_{\hat{x_i}} p_{LM}(\hat{x_i}|x_{\setminus i}) f_y(x_{\setminus i} \cup \hat{x_i}) \tag{2}$$

where, $f_y(x_{\setminus i} \cup \hat{x_i})$ is the prediction of the classification model after the language model's prediction $\hat{x_i}$ is added to the incomplete input $x_{\setminus i}$.

The other method OLM-S calculates the sensitivity of a position in the text and has nothing to do with the word present at that position in the original input. The sensitivity score of OLM-S is calculated

$$s_{f,y}(x_i) = \sqrt{\textstyle\sum_{\hat{x_i}} p_{LM}(\hat{x_i}|x_{\setminus i})(f_y(x_{\setminus i} \cup \hat{x_i}) - \mu)^2}$$

where μ is the mean value from Equation 2. The sensitivity score $s_{f,y}(x_i)$ is used for word ranking in OLM-S.

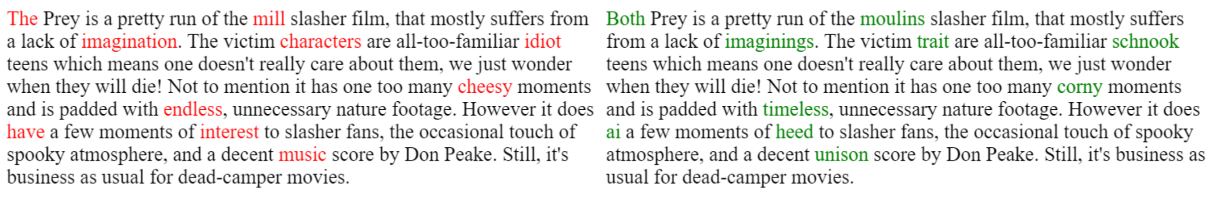
After performing the Word Ranking step using the relevance scores generated by OLM and OLM-S, the next step is to replace highly scored words with semantically similar words that form grammatically correct sentences (Word Replacement) such that the output prediction changes. Word replacement strategy is kept similar to existing methods. TextFooler uses Synonym Extraction, POS checking and semantic similarity checking whereas BAE-R uses a Language Model for word replacement. (details in Appendix C).

Dataset	Classes	Train	Test	Avg. Length
Classification				
IMDB	2	25K	500	245.12
Yelp	2	560K	500	132.33
AG's News	4	120K	500	40.41
Natural Language Inference				
MNLI	3	433K	500	29.72

Table 1: Statistic of Datasets. Avg. Length is the average number of words in the test set.

4 Experiments

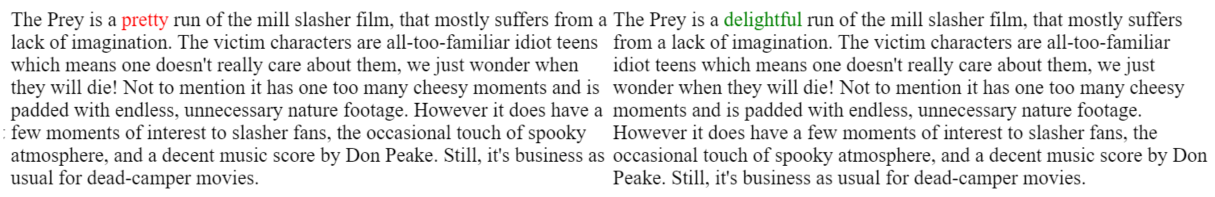
We experiment with different benchmark datasets for text classification and entailment: IMDB, AG News, Yelp Polarity and MNLI (details in Ap-



The Prey is a pretty run of the mill slasher film, that mostly suffers from a lack of imagination. The victim characters are all-too-familiar idiot teens which means one doesn't really care about them, we just wonder when they will die! Not to mention it has one too many cheesy moments and is padded with endless, unnecessary nature footage. However it does have a few moments of interest to slasher fans, the occasional touch of spooky atmosphere, and a decent music score by Don Peake. Still, it's business as usual for dead-camper movies.

Both Prey is a pretty run of the moulins slasher film, that mostly suffers from a lack of imaginings. The victim trait are all-too-familiar schnook teens which means one doesn't really care about them, we just wonder when they will die! Not to mention it has one too many corny moments and is padded with timeless, unnecessary nature footage. However it does ai a few moments of heed to slasher fans, the occasional touch of spooky atmosphere, and a decent unison score by Don Peake. Still, it's business as usual for dead-camper movies.

(a) TextFooler Attack on fine-tuned BERT on IMDB data sample. $[Negative(100\%) \rightarrow Positive(59\%)]$



The Prey is a pretty run of the mill slasher film, that mostly suffers from a lack of imagination. The victim characters are all-too-familiar idiot teens which means one doesn't really care about them, we just wonder when they will die! Not to mention it has one too many cheesy moments and is padded with endless, unnecessary nature footage. However it does have a few moments of interest to slasher fans, the occasional touch of spooky atmosphere, and a decent music score by Don Peake. Still, it's business as usual for dead-camper movies.

The Prey is a delightful run of the mill slasher film, that mostly suffers from a lack of imagination. The victim characters are all-too-familiar idiot teens which means one doesn't really care about them, we just wonder when they will die! Not to mention it has one too many cheesy moments and is padded with endless, unnecessary nature footage. However it does have a few moments of interest to slasher fans, the occasional touch of spooky atmosphere, and a decent music score by Don Peake. Still, it's business as usual for dead-camper movies.

(b) Adv-OLM Attack on fine-tuned BERT on IMDB data sample. $[Negative(100\%) \rightarrow Positive(95\%)]$

Figure 1: Qualitative Examples of TextFooler and Adv-OLM on BERT classifier (Red words are replaced by Green words while changing the output prediction probability.)

Dataset	**Method**	**Word Ranking**	**BERT**				**ALBERT**			
			Original Acc.	**Attacked Acc.**	**Success Rate**	**Perturbed %**	**Original Acc.**	**Attacked Acc.**	**Success Rate**	**Perturbed %**
AG's News	BAE-R	OLM	93.8%	**78%**	**16.84%**	6.72%	94.4%	**79.2%**	**16.1%**	8.37%
		OLM-S		79%	15.78%	**6.35%**		82.4%	12.71%	**6.9%**
		Original (delete)		78.8%	15.99%	6.42%		79.4%	15.89%	7.67%
	TextFooler	OLM		**19.2%**	**79.53%**	23.52%		20.4%	78.39%	21.24%
		OLM-S		21.4%	77.19%	**20.96%**		**20.2%**	**78.6%**	**20.18%**
		Original (delete)		21.4%	77.19%	23.19%		22.0%	76.69%	21.15%
	PWWS	-		44.8%	52.24%	16.21%		36.8%	61.02%	14.8%
Yelp	BAE-R	OLM	97.2%	**38.4%**	**60.49%**	**6.45%**	97.4%	41.4%	57.49%	7.07%
		OLM-S		55.2%	43.21%	10.36%		39.6%	59.34%	7.11%
		Original (delete)		41.6%	57.20%	7.28%		35.0%	64.07%	6.49%
	TextFooler	OLM		**5.2%**	**94.65%**	**9.10%**		**2.6%**	**97.33%**	10.17%
		OLM-S		7.8%	91.98%	13.31%		2.8%	97.13%	10.13%
		Original (delete)		6.6%	93.21%	9.95%		3.8%	96.1%	9.57%
	PWWS	-		6.2%	93.62%	6.9%		3.8%	96.1%	6.82%

Table 2: Comparison between word ranking strategies on AG's News and Yelp for fine-tuned BERT and ALBERT. Our method Adv-OLM has OLM (or OLM-S) as the word ranking strategy.

Dataset	**Method**	**Word Ranking**	**Original Acc.**	**Attacked Acc.**	**Success Rate**	**Perturbed %**
MNLI	BAE-R	OLM	84.6%	20.6%	75.65%	7.82%
		OLM-S		20.8%	75.41%	8.22%
		Original (delete)		14.0%	83.45%	6.4%
	TextFooler	OLM		**6.6%**	**92.2%**	8.29%
		OLM-S		7.0%	91.73%	8.59%
		Original (delete)		6.8%	91.96%	6.98%
	PWWS	-		3.2%	96.22%	6.62%

Table 3: Comparsion between previous methods and Adv-OLM on MNLI fine-tuned BERT. Our method Adv-OLM has OLM (or OLM-S) as the word ranking strategy.

pendix A). The statistics of the final dataset are shown in Table 1. Test set was randomly choosen stratified set. For evaluating the effectiveness of our proposed approach, we experiment with SOTA text classifiers i.e. transformer based models like BERT (Devlin et al., 2018), ALBERT (Lan et al., 2019), RoBERTa (Liu et al., 2019) and DistilBERT (Sanh et al., 2019).

We replaced the existing word ranking strategies (i.e. Original (delete)) of previous attack methods: Textfooler (Jin et al., 2019) and BAE-R (Garg and Ramakrishnan, 2020) with word rankings generated using OLM and OLM-S while keeping rest of the attack procedure same. The

Model	Method	Word Ranking	Original Acc.	Attacked Acc.	Success Rate	Perturbed %
BERT	BAE-R	OLM	92.0%	38.8%	57.83%	2.79%
		OLM-S		**33.2%**	**63.91%**	**2.33%**
		Original (delete)		42.6%	53.70%	2.92%
	Textfooler	OLM		29.8%	67.61%	4.52%
		OLM-S		**26.4%**	**71.3%**	**3.16%**
		Original (delete)		31.8%	65.43%	5.26%
	PWWS	-		28.2%	69.35%	2.92%
ALBERT	BAE-R	OLM	92.8%	27.4%	70.47%	3.58%
		OLM-S		**23.2%**	**75.0%**	**3.4%**
		Original (delete)		26.8%	71.12%	3.48%
	Textfooler	OLM		**1.4%**	**98.49%**	6.65%
		OLM-S		**1.4%**	**98.49%**	**5.39%**
		Original (delete)		2.4%	97.41%	6.57%
	PWWS	-		3.8%	95.91%	3.76%
RoBERTa	BAE-R	OLM	94.2%	29.4%	68.79%	4.08%
		OLM-S		**28.0%**	**70.28%**	**3.57%**
		Original (delete)		29.4%	68.79%	3.9%
	Textfooler	OLM		**0.0%**	**100%**	7.62%
		OLM-S		0.2%	99.79%	**6.43%**
		Original (delete)		0.2%	99.79%	6.89%
	PWWS	-		0.4%	99.58%	5.38%
DistilBERT	BAE-R	OLM	91.8%	22.6%	75.38%	3.29%
		OLM-S		21.6%	76.47%	**2.93%**
		Original (delete)		21.6%	76.47%	3.34%
	Textfooler	OLM		0.2%	99.78%	4.03%
		OLM-S		0.2%	99.78%	**3.55%**
		Original (delete)		0.2%	99.78%	4.44%
	PWWS	-		0.6%	99.35%	3.0%
BiLSTM	BAE-R	OLM	83.8%	10.8%	87.11%	2.78%
		OLM-S		10.4%	87.59%	2.71%
		Original (delete)		8.6%	89.74%	2.52%
	Textfooler	OLM		0%	100%	2.38%
		OLM-S		0%	100%	2.41%
		Original (delete)		0%	100%	1.95%
	PWWS	-		0%	100%	1.63%

Table 4: Comparison between previous methods and Adv-OLM for IMDB dataset across different models. Our method Adv-OLM has OLM (or OLM-S) as the word ranking strategy.

comparison is provided between the attacks generated through original word ranking, and OLM adapted word ranking (including comparison with PWWS attack method (Ren et al., 2019)) in table 2, table 3 and table 4. PWWS (Probability Weighted Word Saliency) method considers the word saliency along with the classification probability. The change in value of the classification probability is used to measure the attack effect of the proposed substitute word, while word saliency shows how well the original word affects the classification. We use the default language model (BERT) employed in the OLM and OLM-S, and kept the number of samples generated by the OLM language model as 30 in all the experiments.

The following evaluation metrics are used:

- **Attacked Acc.**: Accuracy of the model after attack. Lower the better.
- **Success Rate:** Ratio of number of successful attacks and the total number of attempted attacks[1]. Higher the better.
- **Perturbed Percentage:** Ratio of number of words that were modified by the attack and the total number of words in the input example. Lower the better.

We use TextAttack's (Morris et al., 2020) fine tuned models on these datasets and used it to execute the attacks, including Adv-OLM (Appendix B).

Number of queries in Adv-OLM: From equations 2 and 3, it is clear that unlike other methods of deletion and *[UNK]* token replacement, which perform only a single query, we need to perform multiple queries. We set the number of samples

[1]Note that total number of attempted attacks are not the same as number of input examples i.e., the samples which were originally wrongly classified by the model even before an attack are skipped

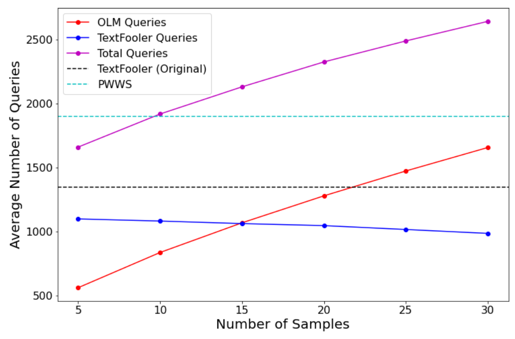


Figure 2: Average Number of Queries vs Number of samples in Adv-OLM attack on BERT on IMDB dataset.

generated by the OLM language model to 30 for our experiment. In the worst case, we would have all 30 samples of the token as unique, which will query the model 30 times. However, experimentally it was not the case. To study this, we varied the number of samples and evaluated the OLM ranking step's number of queries. In fig 2, we plotted the number of queries for OLM averaged over the input samples against the number of samples. We can see that there is not a significant difference in the total number of samples in our case (OLM + Textfooler queries) when compared with PWWS.

5 Results and Analysis

Results are shown in Tables 2, 3 and 4. Table 2 provides the results on AG News and Yelp datasets on fine-tuned BERT and ALBERT model. Our method performs better on both datasets by increasing the success rate by about 1-3% than the previous methods and also decreasing the perturbation percentage. Table 3 gives the results of attacking a fine-tuned BERT on MNLI. Although we did not perform better than original BAE-R, we were still able to outperform TextFooler. Due to the unavailability of MNLI fine-tuned ALBERT model in TextAttack, we did not perform an attack on ALBERT. It can also be seen from Table 2 that the perturbation percentage for AG's News exceed more than 20%, which seems to be a perceptible change, but since the average length of the article is only 40.41, making the space for finding relevant words less, the perturbation percentage becomes very high.

To compare attacks across different transformer-based models, we evaluate the performance of Adv-OLM on IMDB dataset. Table 4 provides the results of different attack methods on BERT, ALBERT, RoBERTa, DistilBERT and BiLSTM. Adv-OLM was able to outperform previous attack methods on BERT, ALBERT, RoBERTa by increasing the success rate up to 10% for BAE-R and up to 6% for TextFooler. Perturbation percentage was also reduced by 1-2%. On DistilBERT, Adv-OLM showed no change in the success rate, but the perturbation percentage was lowered slightly. We also performed an attack on a non-transformer based BiLSTM model which did not show any improvements in the success rate. For BAE-R, it even showed a decrease in the success rate for Adv-OLM. One possible reason for this might be that in both OLM and OLM-S word sampling is performed using a transformer-based BERT language model. We also have qualitative results on IMDB dataset (Figure 1a, 1b).

Experimentally it was observed that better words were ranked when OLM/OLM-S was used as the Word Ranking strategy (Figure 1b). When comparing with the original methods, Adv-OLM has more number of queries, which is due to the fact that for word rankings, OLM/OLM-S queries the model a number of times, thus increasing the overall queries. However, the difference in the number of Adv-OLM queries with the existing attacking methods is not very significant since the model is queried only for unique words from the samples generated from the language model.

6 Conclusion

In this work, we present **Adv-OLM**, a black box attacking method that uses OLM based word ranking strategy, improving the attack performance significantly over previous methods. We also studied how replacing a single variable in a complex system with a new existing method can improve upon the previously existing attack strategies. For future work, we would like to experiment with other language models in the OLM algorithm. We plan to study the effect of using different transformers for the language model and the target model.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*.

Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. 2020. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In *AAAI*, pages 3601–3608.

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*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751*.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. *arXiv preprint arXiv:1804.07781*.

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.

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970*.

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

David Harbecke and Christoph Alt. 2020. Considering likelihood in nlp classification explanations with occlusion and language modeling. *arXiv preprint arXiv:2004.09890*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. *arXiv preprint arXiv:1909.00986*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*.

Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. 2018. Adversarial examples for natural language classification problems.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

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*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*.

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*.

John X. 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.

Danish Pruthi, Bhuwan Dhingra, and Zachary C Lipton. 2019. Combating adversarial misspellings with robust word recognition. *arXiv preprint arXiv:1905.11268*.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1085–1097.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Samson Tan, Shafiq Joty, Min-Yen Kan, and Richard Socher. 2020. It's morphin' time! Combating linguistic discrimination with inflectional perturbations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2920–2935, Online. Association for Computational Linguistics.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. *arXiv preprint arXiv:1908.07125*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Xiaosen Wang, Hao Jin, and Kun He. 2019. Natural language adversarial attacks and defenses in word level. *arXiv preprint arXiv:1909.06723*.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080.

Appendix

A Datasets

We evaluate our adversarial attacks on text classification and natural language inference datasets. We evaluate our method on 500 samples randomly selected from the test set of the given dataset.

Text classification We used the following text classification datasets:

- **IMDB:** Document-level large Movie Review dataset for binary sentiment classification. [2]
- **Yelp:** The Yelp reviews dataset consists of reviews from Yelp. This is a dataset for sentiment classfication. It is extracted from the Yelp Dataset Challenge 2015 data. [3]
- **AG's News:** Sentence level news-type classification dataset, containing 4 types of news: World, Sports, Business, and Science. [4]

Natural Language Inference

- **MNLI:** The corpus of sentence pairs manually labeled for classification with the labels entailment, contradiction, and neutral, supporting the task of natural language inference (NLI). Unlike SNLI, MNLI is more diverse, based on multi-genre texts, covering transcribed speech, popular fiction, and government reports. [5]

Average Length is the average number of words in the randomly chosen 500 samples taken from its test set for each dataset.

B Textattack

TextAttack is an open-source python framework for adversarial attacks, data augmentation and adversarial training in NLP.

Because of the modularity that TextAttack provides, it enables researchers to construct new attacks from a combination of novel and existing approaches or perform analysis on the already existing approaches. This helps in composing and comparing the attacks in a shared environment. TextAttack makes it easy to perform benchmark comparisons across all the previous attacks performed across models. Text Attack provides clean, readable implementations of 16 adversarial attacks from the literature. Out of which two are sequence to sequence attacks and nine are classification based attacks from the GLUE benchmark. A list of these attacks is presented in Table 5. TextAttack is directly integrated with HuggingFace's transformers and NLP libraries. This allows users to test attacks on models and datasets.

Attack Recipes	
BAE (Garg and Ramakrishnan, 2020)	PWWS (Ren et al., 2019)
Bert-Attack (Li et al., 2020)	TextFooler (Jin et al., 2019)
DeepwordBug (Gao et al., 2018)	HotFlip (Ebrahimi et al., 2017)
Alzantot (Alzantot et al., 2018)	Morpheus (Tan et al., 2020)
IGA (Wang et al., 2019)	Pruthi (Pruthi et al., 2019)
Input-Reduction (Feng et al., 2018)	PSO (Zang et al., 2020)
Seq2Sick (Cheng et al., 2020)	TextBugger (Li et al., 2018)
Kuleshov (Kuleshov et al., 2018)	Fast Alzantot (Jia et al., 2019)

Table 5: Adversarial attacks implemented in Textattack

TextAttack builds attacks from four components:

1. A **search method** that selects the words to be transformed.
2. A **transformation** that generates a set of possible perturbations for the given input.
3. A **set of constraints** implied on the transformation to ensure that the perturbations are valid with respect to the original input.
4. A **goal function** that determines whether an attack is successful in terms of model outputs. For classification tasks, untargeted, and targeted. For a sequence to sequence tasks, non-overlapping output, and minimum BLEU score.

For our approach, we attack TextAttack's fine-tuned models on datasets discussed in A, that are publically available on huggingface[6] Textattack is also used for the execution of all the previous attacking methods and our Adv-OLM as well.

C Word Replacement Strategies

C.1 TextFooler Word Replacement Strategy

Following workflow was proposed by the paper:

Synonym Extraction: Gather a candidate set CANDIDATES for all possible replacements of the selected word w_i and every other word in the vocabulary. To represent the words, counter fitting word embeddings were used. Using this set

[2]IMDB dataset
[3]Yelp dataset
[4]AG's News dataset
[5]MNLI dataset

[6]TextAttack fine-tuned models on HuggingFace

of embedding vectors, top N synonyms whose cosine similarity with w is higher than some δ were chosen.

POS Checking: In the set CANDIDATES of the word w_i, only the ones with the same part-of-speech(POS) as w_i were kept. This step assures that the grammar of the text is mostly maintained.

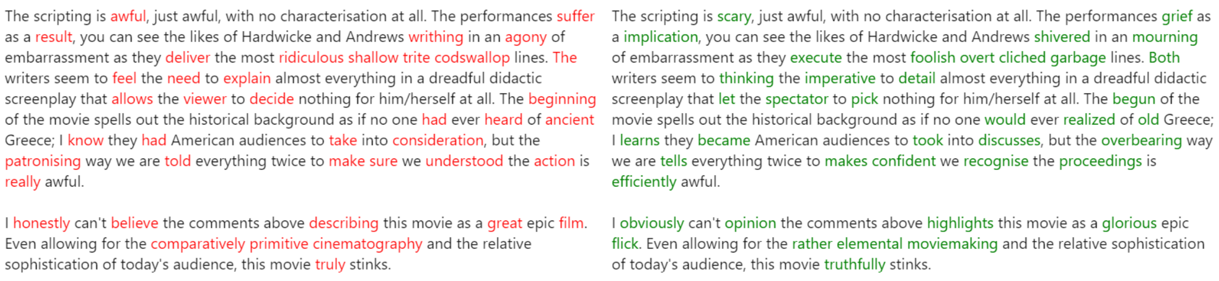
Semantic Similarity Checking: For each remaining word $c \in$ CANDIDATES, these were substituted for w_i in the sentence X, and an adversarial example X_{adv} was obtained. Universal Sentence Encoder (USE) was used to encode the two sentences into high dimensional vectors and then use their cosine similarity score to calculate the sentence similarity between X and X_{adv}. The words resulting in similarity scores above a preset threshold ϵ were placed in a final candidate pool (FINCANDIDATE).

Finally, every candidate word from the FINCANDIDATE was chosen one by one, and the one that resulted in the least confidence score of label y was considered as the best replacement for word w_i.

C.2 BAE-R Word Replacement Strategy

BAE uses a pre-trained BERT masked language model(MLM) to predict the mask tokens for replacement. Since BERT is powerful and trained on the large training corpus, the predicted mask tokens fit well grammatically in the sentence. BERT-MLM does not, however, guarantee semantic coherence to the original text. To ensure semantic similarity on introducing perturbations in the input text, a set of K masked tokens were filtered out using Universal Sentence Encoder(USE) based on sentence similarity score. An additional check for grammatical correctness of the generated adversarial example by filtering out predicted tokens that do not form the same part of speech(POS) as the original token in the sentence was performed.

D More Examples

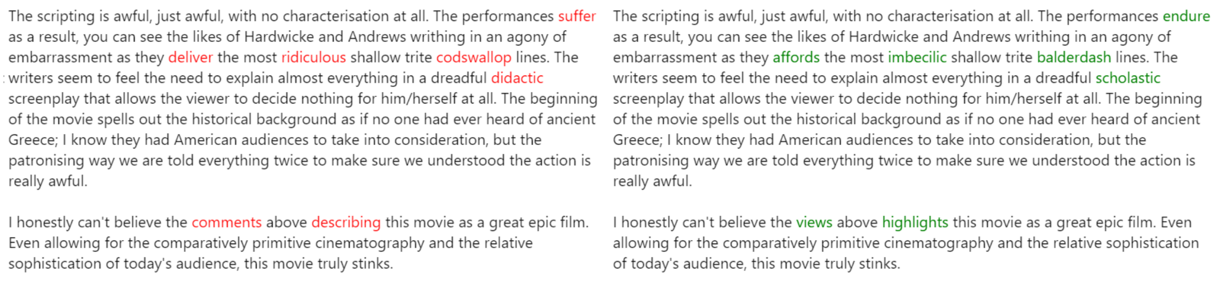

The scripting is awful, just awful, with no characterisation at all. The performances suffer as a result, you can see the likes of Hardwicke and Andrews writhing in an agony of embarrassment as they deliver the most ridiculous shallow trite codswallop lines. The writers seem to feel the need to explain almost everything in a dreadful didactic screenplay that allows the viewer to decide nothing for him/herself at all. The beginning of the movie spells out the historical background as if no one had ever heard of ancient Greece; I know they had American audiences to take into consideration, but the patronising way we are told everything twice to make sure we understood the action is really awful.

I honestly can't believe the comments above describing this movie as a great epic film. Even allowing for the comparatively primitive cinematography and the relative sophistication of today's audience, this movie truly stinks.

The scripting is scary, just awful, with no characterisation at all. The performances grief as a implication, you can see the likes of Hardwicke and Andrews shivered in an mourning of embarrassment as they execute the most foolish overt cliched garbage lines. Both writers seem to thinking the imperative to detail almost everything in a dreadful didactic screenplay that let the spectator to pick nothing for him/herself at all. The begun of the movie spells out the historical background as if no one would ever realized of old Greece; I learns they became American audiences to took into discusses, but the overbearing way we are tells everything twice to makes confident we recognise the proceedings is efficiently awful.

I obviously can't opinion the comments above highlights this movie as a glorious epic flick. Even allowing for the rather elemental moviemaking and the relative sophistication of today's audience, this movie truthfully stinks.

(a) TextFooler Attack on fine-tuned ALBERT on IMDB data sample. [$Negative(100\%) \rightarrow Positive(51\%)$]


The scripting is awful, just awful, with no characterisation at all. The performances suffer as a result, you can see the likes of Hardwicke and Andrews writhing in an agony of embarrassment as they deliver the most ridiculous shallow trite codswallop lines. The writers seem to feel the need to explain almost everything in a dreadful didactic screenplay that allows the viewer to decide nothing for him/herself at all. The beginning of the movie spells out the historical background as if no one had ever heard of ancient Greece; I know they had American audiences to take into consideration, but the patronising way we are told everything twice to make sure we understood the action is really awful.

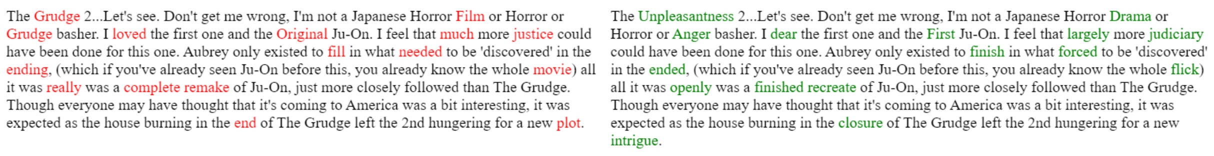
I honestly can't believe the comments above describing this movie as a great epic film. Even allowing for the comparatively primitive cinematography and the relative sophistication of today's audience, this movie truly stinks.

The scripting is awful, just awful, with no characterisation at all. The performances endure as a result, you can see the likes of Hardwicke and Andrews writhing in an agony of embarrassment as they affords the most imbecilic shallow trite balderdash lines. The writers seem to feel the need to explain almost everything in a dreadful scholastic screenplay that allows the viewer to decide nothing for him/herself at all. The beginning of the movie spells out the historical background as if no one had ever heard of ancient Greece; I know they had American audiences to take into consideration, but the patronising way we are told everything twice to make sure we understood the action is really awful.

I honestly can't believe the views above highlights this movie as a great epic film. Even allowing for the comparatively primitive cinematography and the relative sophistication of today's audience, this movie truly stinks.

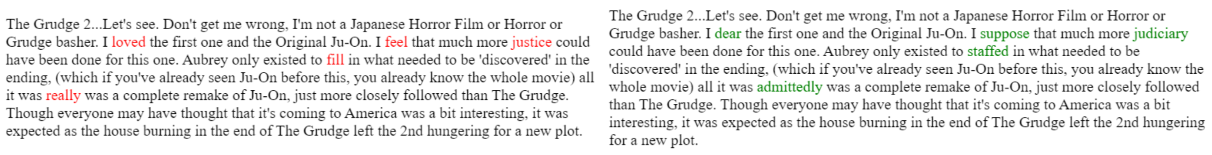
(b) Adv-OLM attack on fine-tuned ALBERT on IMDB data sample. [$Negative(100\%) \rightarrow Positive(57\%)$]

Figure 3: Qualitative Examples of TextFooler and Adv-OLM on ALBERT classifier (Red words are replaced by Green words while changing the output prediction probability.)


The Grudge 2...Let's see. Don't get me wrong, I'm not a Japanese Horror Film or Horror or Grudge basher. I loved the first one and the Original Ju-On. I feel that much more justice could have been done for this one. Aubrey only existed to fill in what needed to be 'discovered' in the ending, (which if you've already seen Ju-On before this, you already know the whole movie) all it was really was a complete remake of Ju-On, just more closely followed than The Grudge. Though everyone may have thought that it's coming to America was a bit interesting, it was expected as the house burning in the end of The Grudge left the 2nd hungering for a new plot.

The Unpleasantness 2...Let's see. Don't get me wrong, I'm not a Japanese Horror Drama or Horror or Anger basher. I dear the first one and the First Ju-On. I feel that largely more judiciary could have been done for this one. Aubrey only existed to finish in what forced to be 'discovered' in the ended, (which if you've already seen Ju-On before this, you already know the whole flick) all it was openly was a finished recreate of Ju-On, just more closely followed than The Grudge. Though everyone may have thought that it's coming to America was a bit interesting, it was expected as the house burning in the closure of The Grudge left the 2nd hungering for a new intrigue.

(a) TextFooler Attack on fine-tuned ALBERT on IMDB data sample. [$Negative(100\%) \rightarrow Positive(51\%)$]


The Grudge 2...Let's see. Don't get me wrong, I'm not a Japanese Horror Film or Horror or Grudge basher. I loved the first one and the Original Ju-On. I feel that much more justice could have been done for this one. Aubrey only existed to fill in what needed to be 'discovered' in the ending, (which if you've already seen Ju-On before this, you already know the whole movie) all it was really was a complete remake of Ju-On, just more closely followed than The Grudge. Though everyone may have thought that it's coming to America was a bit interesting, it was expected as the house burning in the end of The Grudge left the 2nd hungering for a new plot.

The Grudge 2...Let's see. Don't get me wrong, I'm not a Japanese Horror Film or Horror or Grudge basher. I dear the first one and the Original Ju-On. I suppose that much more judiciary could have been done for this one. Aubrey only existed to staffed in what needed to be 'discovered' in the ending, (which if you've already seen Ju-On before this, you already know the whole movie) all it was admittedly was a complete remake of Ju-On, just more closely followed than The Grudge. Though everyone may have thought that it's coming to America was a bit interesting, it was expected as the house burning in the end of The Grudge left the 2nd hungering for a new plot.

(b) Adv-OLM attack on fine-tuned ALBERT on IMDB data sample. [$Negative(100\%) \rightarrow Positive(65\%)$]

Figure 4: Qualitative Examples of TextFooler and Adv-OLM on RoBERTa classifier (Red words are replaced by Green words while changing the output prediction probability.)