# Computationally Efficient Gradient Based Whitebox Adversarial Attack against Text Transformers

**Khushbu Pahwa**
Department of Computer Science
University of California, Los Angeles
khushbu16pahwa@g.ucla.edu

**Jakob Didio**
Department of Computer Science
University of California, Los Angeles
jakobdidio@ucla.edu

**Hsin-Wei Yu**
Department of Computer Science
University of California, Los Angeles
hsinweiyu@ucla.edu

## Abstract

Deep learning models are now widely deployed in safety critical, medical, and other security demanding application environments, and are becoming highly ubiquitous. Thus there is an ever-increasing concern about the vulnerabilities of these models and their susceptibility to adversarial attacks. Most of the adversarial attacks studied in prior literature have focused on computer vision applications because the perturbation space is continuous, and thus gradient based optimization problems can be solved. However, extending this to text is not trivial because of the discrete attack space. In this paper, we attempt to reduce the computational complexity involved in gradient based whitebox attack against a gpt2 model pre-trained on the downstream NLP tasks of news categorization of the AG News dataset, and natural language inference on the MNLI dataset. Particulary, we attempt to attain performance increases in attacking the text samples while maintaining a high average semantic similarity in the adversarial output text, along with a high degree of imperceptibility.

## 1 Introduction

Adversarial attacks on deep neural networks (DNNs) tend to add slight noise to the inputs of DNNs and perturb the models to make wrong predictions. [1] There are different methods for constructing adversarial examples, including optimized-based search strategies and gradient-based adversarial attacks. The former method encourages prediction error and minimizes the adversarial loss. This method introduces a perceptibility constraint into the optimization problem. Previous works [2, 3] have been successfully applied to image and speech data. However, it is more challenging to obtain adversarial examples with text data for optimization-based search strategies.

Natural languages have an inherently discrete nature that makes gradient descent techniques utilized in attacks against continuous data like images not directly applicable to attacking language models. Some works [4, 5, 6] leverage heuristic word replacement strategies by greedy or beam search using black-box queries. However, the changes made by these strategies are often unnatural and nonsensical. In [7], the authors propose a general-purpose framework for gradient-based adversarial attacks and applied it against transformer models on text data. Their framework, Gradient-Based Distributional Attack (GBDA), contains components that convert the task of attacking a transformer model into one that can be optimized with gradient-based methods, and also introduce perceptibility constraints in order to reduce perceptibility of adversarial examples.

However, there are several limitations to GBDA. First, the perturbations supported by GBDA are limited to token replacements, while insertions or deletions would also be desirable modifications. Second, the adversarial distribution $\theta$ is highly over-parameterized. (i.e., $\theta$ is of size $n \times V$, where $n$ is the number of tokens for a given example and $V$ is vocabulary size) This may cause issues with space (e.g., out of memory) or attack duration (e.g., 10 seconds per sentence) for sentences that are extremely long.

Due to the motivation above, we consider methods to reduce the size of $\theta$ or avoid reliance on the over-parameterized distribution at all while achieving high semantic similarity. Simultaneously, we aim to reduce time needed to attack and obtain a higher degree of imperceptibility.

The report is organized as follows: Section II talks about the related works in the field that have investigated whitebox attack scenarios for NLP tasks. Section III introduces the Gradient Based Distributional Attack. Section IV describes our approach with synonym replacement to reduce the vocabulary for perturbing the sample inputs[1], and sparsity loss and projecting the gradient of embeddings to the the most close semantically similar token in the discrete vocabulary space. Section V details our experimental setup. Finally, Section VI serves as our experimental results discussion and conclusion.

## 2 Related Work

Due to the challenges in optimization and perceptibility associated with the discrete nature of textual data, very few works exist that explore white-box attacks against NLP models. A common approach for applying gradient based strategies is to compute them in the continuous gradient space. The common strategies proposed in related works are random word replacement in the input sentence by the nearest word in the embedding space with the smallest difference with the original word in the direction of the gradient [8]. There is another work [9] that extends Adv-Text [10] to generate adversarial perturbations by imposing the direction of the perturbations in the embedding space to align with meaningful embedding vectors. The major drawback of these strategies is that the generated adversarial text successfully fools the model but it is extremely perceptible. However, Facebook AI Research [7] has proposed Gradient-based Distributional Attack (GBDA) against text transformers. They consider a probability distribution over an entire vocabulary for each word in the adversarial sentence. They optimize this continuous matrix to cause the target model to misclassify.

## 3 GBDA attack

Facebook AI Research introduced Gradient Based Adversarial Attack (GBDA) for attacking text transformers. They create an adversarial distribution which can be sampled to produce adversarial examples. $\Theta$ is a Gumbel-Softmax distribution, which is continuous and differentiable Adversarial loss is augmented with fluency and similarity constraints, both of which are differentiable BERTScore similarity constraint is used to measure the semantic similarity of the pairwise tokens. Causal Language Model determines the fluency constraint.They augment the adversarial loss with fluency constraints and the semantic similarity constraints, thus making their white-box attack produces more natural adversarial texts. However, there are some key limitations of their work that they address in their paper: 1) Their attack accounts for only token replacement and not insertions and deletions, thus the adversarial text output lacks the naturalness that one would expect to claim that the generated adversarial text is imperceptible. 2) The adversarial distribution matrix $\Theta$ is highly over-parameterized and might be a big bottleneck with regards to space and time complexity when dealing with long text sequences.

## 4 Our Approaches

This discussion will cover two approaches to overcome the limitations of the GBDA attack detailed in section 3. With our approaches, we intend to specifically address the limitation of GBDA attack regarding its utilization of a very over-parameterized probability distribution. Thus, we explore approaches that can achieve speedup in attack time and enhance imperceptibility in the generated output while not compromising on the attack success rate.

[1]Code is publicly available here: https://github.com/kpahwa16/CS269_Final_Project

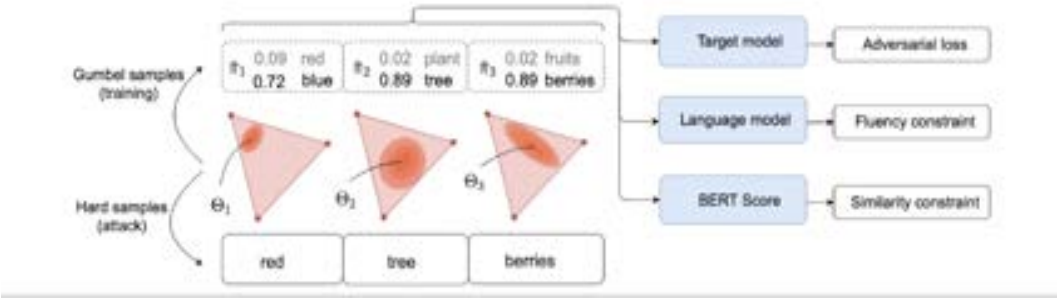


Figure 1: Architecture of GBDA

### 4.1 Synonym Parameterization

When considering ways to perturb a target sentence while preserving imperceptibility, we made the assumption that replacing tokens in a sentence with their synonyms would more closely preserve the original semantics of the sentence. Additionally, when considering the task of reducing the overparameterization of $\Theta$, it is viable to reduce either the size of $n$, being the number of tokens considered for replacement in the original sentence, or $V$, being the vocabulary size that the perturbations are drawn from. By adopting the synonym replacement strategy intuition described above, a result would be decreasing the size of $V$, as the total number of synonyms for the tokens in a given sentence is likely to be much smaller than an arbitrary vocabulary. However, obtaining the synonyms for tokens in an arbitrary sentence is a difficult task to perform precisely, as punctuation, plurality, capitalization, and different word meanings can result in irrelevant words being fetched. For instance, the word "dump" can be a verb meaning to place garbage in an area, or it can be a noun representing an area where garbage is often placed. For the approach taken in this work, we first pre-process an input sentence using standard heuristic cleaning strategies to remove capitals, punctuation, etc. and then consult a synonym database for each word. We select the first "sense" of a word given by the synonym database to gather related words. A clear improvement that could be made here is to use a more comprehensive model for determining the proper meaning of a word in context, and using that to select a proper collection of synonyms.

**Algorithm 1:** Synonym Distribution Algorithm

1: **Input:**
   token sequence $\{a_1, a_2, \ldots, a_n\}$
2: For given token sequence, process to remove punctuation and make all characters lowercase using nltk[2]
3: Retrieve synonyms for the first "sense" of each token using the nltk wordnet database.
4: Concatenate synonyms for all tokens together into one string
5: Use the pretrained GPT2 model fine-tuned on the target dataset to retrieve the embeddings for the resulting string. (Original $\Theta$ is of size $n \times$ V, new $\Theta$ is of size n x K, where K is # of synonyms for all tokens and K $\leq$ V).
6: Conduct GBDA using the newly constructed $\Theta$

### 4.2 Gradient Projection Attack with L2L1 Sparsity

In this approach, we attempt to maximize imperceptibility while ensuring semantic similarity in the output adversarial text. Thus, only a few tokens should be changed for imperceptibility and only a few blocks of the perturbation vector should be nonzero. Therefore, we add the (L2L1) loss term for the perturbation vector in the optimization problem, which is also referred to as the Block Sparsity Loss in the non-convex optimization literature. While initially exploring ideas for the project, our team discussed this idea, and were experimenting with it. Later while conducting a literature survey, we realized that a similar approach has been proposed in [11]. The approach used in the paper is not novel since there are many works that talk about gradient projection and augmenting the adversarial loss with some sparsity constraint like [12] that proposed a projected

gradient method combined with group lasso and gradient regularization. By changing less than 3 words, seq2seq models could produce desired outputs with high success rates. We originally derived the motivation for our approach from the Seq2Sick paper. Thus, our contribution here is to evaluate different approaches with respect to the attack time, attack success rate, imperceptibility measure in terms of average semantic similarity in successful attacks and the token error rate. The algorithm is defined below:

Consider f : X $\rightarrow$ Y to be the target text classifier model which correctly predicts the class of the input sentence x $\in$ X to be y = f(x) $\in$ Y. Each input sample is expressed as a sequence of tokens belonging to the vocabulary set V. The tokens of these sentences are in a discrete space. Therefore, each of these tokens is transformed to a continuous vector, called an embedding vector, as the input of the target transformer model. Let emb(.) denote the embedding function that gets a token as the intput and transforms it to a continuous vector. Therefore, we can represent the sentence x as a sequence of tokens $[t_1, t_2, .....t_n]$, and the the embedding space as a sequence of embedding vectors $e_x$ = $[emb(x_1), emb(x_2), ..., emb(x_n)]$ by transforming each of its tokens by the function emb(.). Similarly, let $e'_x$ = $e_x + r_x$ represents the adversarial example as a sequence of embedding vectors. $r_x$ = $[r_1, r_2, ..., r_n]$ is the sequence of the perturbation vectors of each token. Now, we augment the adversarial loss function defined as $L_{adv}$ which is the negative of the cross-entropy loss with an L2L1 Block Sparisty Loss. Thus, the complete discrete optimization problem is defined in the following equation:

$$e'_x = \arg\min_{e \in E_v} L_{adv} + \lambda L_{Sparsity} \tag{1}$$

Here, where $E_v$ denotes the discrete subspace of every token of the vocabulary set V in the embedding space. Moreover, $\lambda$ that controls the degree of imperceptibility by weighting the L2L1 sparsity loss defined in the equation below:

$$L_{sparsity} = \sum_{i=1}^{n} ||p_i||^2 \tag{2}$$

Here, $p_i$ denotes the perturbation vector for the $i^{th}$ token. Thus, $L_{sparsity}$ is basically the L1 relaxation of the L2 norms of all the perturbation vectors. Our rationale for this formulation of the sparsity loss as also nicely highlighted in [11] is that non-zero entries of the perturbation $r_x$ occur in clusters, which implies that $r_x$ should be block-sparse. The algorithm is described in the flowchart below (Fig. 2).

In each iteration, the embedding vectors of all the tokens of the adversarial example in the continuous space are updated. These updated embeddings are then projected to the the embedding vectors of the closest meaningful tokens. We use cosine similarity metric to find the closest embedding vectors and apply the projection for each token independently. Since we are dealing with discrete data, it is possible that through iterations we come across a previously computed embedding vector after the projection. Moreover, if the perturbation vector is too small, the updated vectors will be projected to the previous sentence. In these cases, the algorithm will be stuck in a loop as the computed gradients will stay the same. To prevent this undesirable scenario, we update the embedding vectors in the projection step only when the projected sentence has not been generated before. To this end, we save all the updated sentences in a buffer, and update the embedding vectors by the projected ones only if the output of the projection step is not in the buffer. These steps are performed iteratively until the target model is fooled or a maximum number of iterations is reached (the algorithm fails to find an adversarial example in this case).

We discuss the results of this approach in Section 5.

| Evaluation Metrics | GBDA | Gradient Projection (L2L1 sparsity) w/ decreasing $\lambda$ | Gradient Projection (L2L1 sparsity) w/ fixed $\lambda$ |
|---|---|---|---|
| Average Semantic Similarity in Successful Attacks | 0.70 | 0.86 | 0.90 |
| No of failed attacks due to lower similarity | 1 | 0 | 0 |
| Token Error Rate in Successful Attacks | 0.008 | 0.11 | 0.08 |
| Total no. of failed attacks | 4 | 1 | 20 |
| Success Attack Rate | 0.95 | 0.98 | 0.75 |
| Time taken to attack (80 samples) in sec | 1338 | 1098 | 388 |

Table 1: Experimental Results on AG News Dataset

| Evaluation Metrics | GBDA | Gradient Projection (L2L1 sparsity) w/ decreasing $\lambda$ |
|---|---|---|
| Average Semantic Similarity in Successful Attacks | 0.63 | 0.80 |
| No of failed attacks due to lower similarity | 4 | 1 |
| Token Error Rate in Successful Attacks | 0.15 | 0.07 |
| Total no. of failed attacks | 7 | 3 |
| Success Attack Rate | 0.9125 | 0.9625 |
| Time taken to attack (80 samples) in sec | 1395 | 929 |

Table 2: Experimental Results on MNLI Dataset

| Dataset | Sentence | Prediction | Text |
|---|---|---|---|
| GBDA | Original | Business | 'Foreign investors likely to bid for Yukos unit MOSCOW: Foreign investors may take part in the sale of assets in Russian oil major Yukos main production unit, which could be offered at a 60 price discount to settle back taxes, Russian television reported on Monday.' |
| | Adversarial | SciFi/Tech | 'Foreign investors likely to venture on Yukos LP MOSCASE: Foreign investors may take part in the sale of assets in Cyber distribution major SXos main operating unit, which will be used at a lower price tag to pay the debt, Russian media reported on Monday.' |
| Gradient Projection + L2L1 sparsity) | Original | World | 'Yemen Sentences 15 Militants on Terror Charges A court, Yemen has sentenced one man in death and 14 others to prison terms for a series of attacks and terrorist plots in 2002, including the bombing of a French oil tanker.' |
| | Adversarial | Sports | 'Yemen Sentences 15 Militants on Terror Charges A court in Yemen has sentenced one man to death and 14 others to prison terms for a series of attacks and NASCAR plots in 2002, including the bombing of a French oil tanker.' |

Table 3: Adversarial Text outputs by GBDA attack & Gradient Projection Attack on AG NEWS dataset

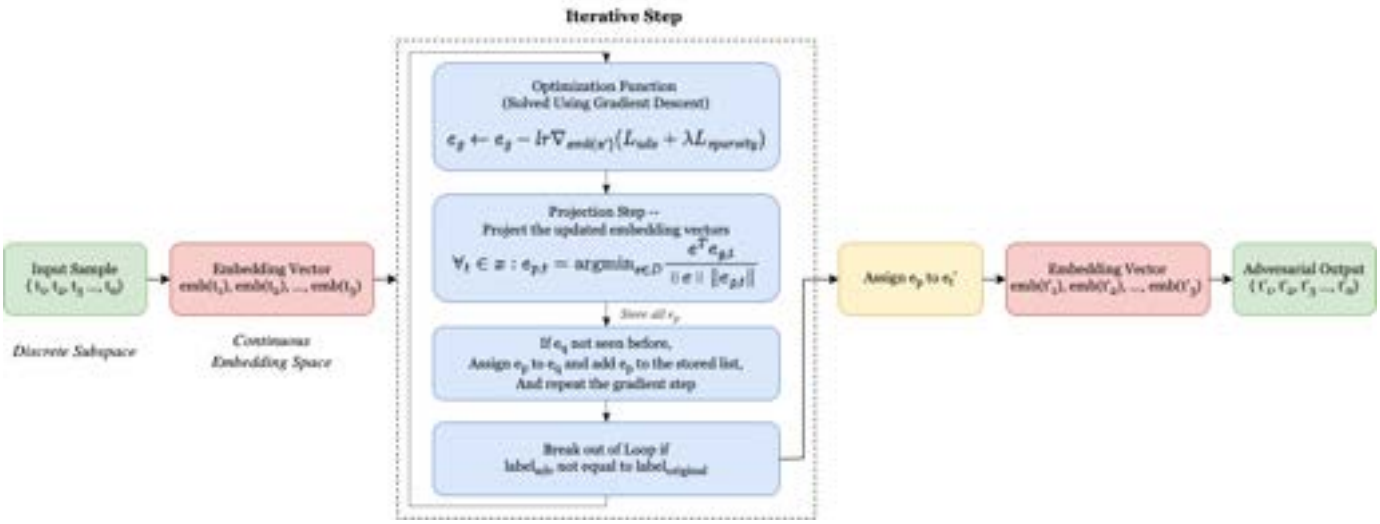


Figure 2: Flowchart of gradient project attack with L2L1 sparsity

## 5 Experimental Setup

### 5.1 Model & Dataset

In this study, we attempt to adversarially attack the GPT2 transformer model [13] pretrained on the news categorization task (AG-NEWS) and the natural language inference task (MNLI). While we show the experimental results for these two tasks, the proposed methodology is effective in attacking the transformer models pre-trained on other downstream NLP tasks as well like sentiment classification, etc.

### 5.2 Metrics

We consider the following evaluation metrics to assess and evaluate the efficacy of the L2L1 sparsity loss + gradient projection approach compared to the baseline GBDA attack: average Semantic Similarity in successful attacks (this is a useful metric that is computed using the cosine similarity on the encoding of the adversarial text and original text obtained by Universal Sentence Encoder), time taken to attack a fixed number of samples (this is a crucial metric as we need to assess if the proposed approach can attain speedup over the GBDA attack), token Error Rate (this metric is important for assessing the imperceptibility of the attack. It measures the word (token) error rate between the original (clean) and the generated adversarial text), and attack Success Rate (this metric is used to validate if the proposed attack can fool the transformer model). Note that we consider a generated adversarial example that has low semantic similarity ($< 0.3$) with the natural example that generated it to be a perceptible adversarial example, and is thus not considered a successful attack.

## 6 Discussion And Conclusion

First, we do not include the results of the synonym generation method in Tables 1 or 2 because although it achieved a high misclassification rate, the generated adversarial examples were extremely perceptible in all cases. We observe from the remaining results that our proposed Gradient Projection with L2L1 sparsity loss beats the performance of GBDA attack in terms of the attack success rate, average semantic similarity in successful attacks, and a lower number of failed attacks. Most importantly, our major contribution here is that the time taken to attack 80 samples for both AG-NEWS and MNLI datasets is reduced considerably compared to the GBDA attack. Thus, we propose a faster attack strategy without the need of any over-parameterization, thus overcoming one of the major drawbacks of the GBDA attack. We propose a simpler strategy with augmenting adversarial loss with L2L1 sparsity loss and using gradient projection to project the embeddings to the most similar tokens in the discrete subspace of the vocabulary.

## References


[1] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013, pp. 387–402.

[2] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[3] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 1–7.

[4] J. Li, S. Ji, T. Du, B. Li, and T. Wang, "Textbugger: Generating adversarial text against real-world applications," *arXiv preprint arXiv:1812.05271*, 2018.

[5] S. Garg and G. Ramakrishnan, "Bae: Bert-based adversarial examples for text classification," *arXiv preprint arXiv:2004.01970*, 2020.

[6] J. X. Morris, E. Lifland, J. Lanchantin, Y. Ji, and Y. Qi, "Reevaluating adversarial examples in natural language," *arXiv preprint arXiv:2004.14174*, 2020.

[7] C. Guo, A. Sablayrolles, H. Jégou, and D. Kiela, "Gradient-based adversarial attacks against text transformers," *arXiv preprint arXiv:2104.13733*, 2021.

[8] N. Papernot, P. McDaniel, A. Swami, and R. Harang, "Crafting adversarial input sequences for recurrent neural networks," in *MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE, 2016, pp. 49–54.

[9] M. Sato, J. Suzuki, H. Shindo, and Y. Matsumoto, "Interpretable adversarial perturbation in input embedding space for text," *arXiv preprint arXiv:1805.02917*, 2018.

[10] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," *arXiv preprint arXiv:1605.07725*, 2016.

[11] S. Sadrizadeh, L. Dolamic, and P. Frossard, "Block-sparse adversarial attack to fool transformer-based text classifiers," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7837–7841.

[12] M. Cheng, J. Yi, P.-Y. Chen, H. Zhang, and C.-J. Hsieh, "Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 3601–3608.

[13] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.