Gradient-based Adversarial Attacks against Text Transformers

Chuan Guo* **Alexandre Sablayrolles*** **Hervé Jégou** **Douwe Kiela**
Facebook AI Research

Abstract

We propose the first general-purpose gradient-based adversarial attack against transformer models. Instead of searching for a single adversarial example, we search for a distribution of adversarial examples parameterized by a continuous-valued matrix, hence enabling gradient-based optimization. We empirically demonstrate that our white-box attack attains state-of-the-art attack performance on a variety of natural language tasks, outperforming prior work in terms of adversarial success rate with matching imperceptibility as per automated and human evaluation. Furthermore, we show that a powerful black-box transfer attack, enabled by sampling from the adversarial distribution, matches or exceeds existing methods, while only requiring hard-label outputs.

1 Introduction

Deep neural networks are sensitive to small, often imperceptible changes in the input, as evidenced by the existence of so-called *adversarial examples* (Biggio et al., 2013; Szegedy et al., 2013). The dominant method for constructing adversarial examples defines an *adversarial loss*, which encourages prediction error, and then minimizes the adversarial loss over the input space with established optimization techniques. To ensure that the perturbation is hard to detect by humans, existing methods also introduce a perceptibility constraint into the optimization problem. Variants of this general strategy have been successfully applied to image and speech data (Madry et al., 2017; Carlini and Wagner, 2017, 2018).

However, optimization-based search strategies for obtaining adversarial examples are much more challenging with text data. Attacks against continuous data types such as image and speech utilize gradient descent for superior efficiency, but the discrete nature of natural languages prohibits such first-order techniques. In addition, perceptibility for continuous data can be approximated with L_2- and L_∞-norms, but such metrics are not readily applicable to text data. To circumvent this issue, some existing attack approaches have opted for heuristic word replacement strategies and optimizing by greedy or beam search using black-box queries (Jin et al., 2020; Li et al., 2020a,b; Garg and Ramakrishnan, 2020). Such heuristic strategies typically introduce unnatural changes that are grammatically or semantically incorrect (Morris et al., 2020a).

In this paper, we propose a general-purpose framework for gradient-based adversarial attacks, and apply it against transformer models on text data. Our framework, **GBDA** (Gradient-based Distributional Attack), consists of two key components that circumvent the difficulties of gradient descent for discrete data under perceptibility constraints. First, instead of constructing a single adversarial example, we search for an *adversarial distribution*. We instantiate examples with the Gumbel-softmax distribution (Jang et al., 2016), parameterized by a continuous-valued matrix of coefficients that we optimize with a vanilla gradient-based method. Second, we enforce perceptibility and fluency using BERTScore (Zhang et al., 2019) and language model perplexity, respectively, both of which are differentiable and can be added to the objective function as soft constraints. The combination of these two components enables powerful, efficient, gradient-based text adversarial attacks.

We empirically demonstrate the efficacy of GBDA against several transformer models. In addition, we also evaluate under the transfer-based black-box threat model by sampling from the optimized adversarial distribution and querying against a different, potentially unknown target model. On a variety of tasks including news/article categorization, sentiment analysis, and natural language inference, our method achieves state-of-the-art attack success rate, while preserving fluency, grammatical

*Equal contribution.

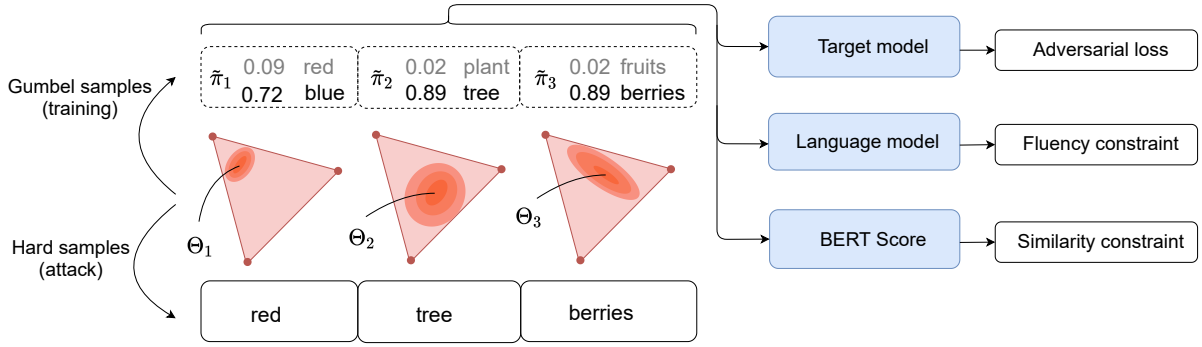

Figure 1: Overview of our attack framework. The parameter matrix Θ is used to sample a sequence of probability vectors $\tilde{\pi}_1, \ldots, \tilde{\pi}_n$, which is forwarded through three (not necessarily distinct) models: (i) the target model for computing the adversarial loss, (ii) the language model for the fluency constraint, and (iii) the BERTScore model for the semantic similarity constraint. Due to the differentiable nature of each loss component and of the Gumbel-softmax distribution, our framework is fully differentiable, hence enabling gradient-based optimization.

correctness, and a high level of semantic similarity to the original input.

In summary, the main contributions of our paper are as follows:

1. We define a parameterized adversarial distribution and optimize it using gradient-based methods. In contrast, most prior work construct a single adversarial example using black-box search.
2. By incorporating differentiable fluency and semantic similarity constraints into the adversarial loss, our white-box attack produces more natural adversarial texts while achieving a new state-of-the-art success rate.
3. The adversarial distribution can be sampled efficiently to query different target models in a black-box setting. This enables a powerful transfer attack that matches or exceeds the performance of existing attacks. Compared to prior work that operate on continuous-valued outputs from the target model, this transfer attack only requires hard labels.

2 Background

Adversarial examples constitute a class of robustness attacks against neural networks. Let $h : \mathcal{X} \to \mathcal{Y}$ be a classifier where $\mathcal{X}, \mathcal{Y}$ are the input and output domains, respectively. Suppose that $\mathbf{x} \in \mathcal{X}$ is a test input that the model correctly predicts as the label $y = h(\mathbf{x}) \in \mathcal{Y}$. An (untargeted) adversarial example is a sample $\mathbf{x}' \in \mathcal{X}$ such that $h(\mathbf{x}') \neq y$ but $\mathbf{x}'$ and $\mathbf{x}$ are imperceptibly close.

The notion of perceptibility is introduced so that $\mathbf{x}'$ preserves the semantic meaning of $\mathbf{x}$ for a human observer. At a high level, $\mathbf{x}'$ constitutes an attack on the model's robustness if a typical human would not misclassify $\mathbf{x}'$ but the model h does. For image data, since the input domain $\mathcal{X}$ is a subset of the Euclidean space $\mathbb{R}^d$, a common surrogate for perceptibility is a distance metric such as the Euclidean distance or the Chebyshev distance. In general, one can define a perceptibility metric $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ and a threshold $\epsilon > 0$ so that $\mathbf{x}'$ is considered imperceptible to $\mathbf{x}$ if $\rho(\mathbf{x}, \mathbf{x}') \leq \epsilon$.

Search problem formulation. The process of finding an adversarial example is typically modeled as an optimization problem. For classification, the model h outputs a *logit vector* $\phi_h(\mathbf{x}) \in \mathbb{R}^K$ such that $y = \arg\max_k \phi_h(\mathbf{x})_k$. To encourage the model to misclassify an input, one can define an *adversarial loss* such as the margin loss:

$$\ell_{\text{margin}}(\mathbf{x}, y; h) = \max\left(\phi_h(\mathbf{x})_y - \max_{k \neq y} \phi_h(\mathbf{x})_k + \kappa, 0\right), \quad (1)$$

so that the model misclassifies $\mathbf{x}$ by a margin of $\kappa > 0$ when the loss is 0. The margin loss has been widely used in attack algorithms for image data (Carlini and Wagner, 2017).

Given an adversarial loss ℓ, the process of constructing an adversarial example can be cast as a constrained optimization problem:

$$\min_{\mathbf{x}' \in \mathcal{X}} \ell(\mathbf{x}', y; h) \quad \text{subject to } \rho(\mathbf{x}, \mathbf{x}') \leq \epsilon. \quad (2)$$

An alternative formulation is to relax the constraint into a soft constraint with $\lambda > 0$:

$$\min_{\mathbf{x}' \in \mathcal{X}} \ell(\mathbf{x}', y; h) + \lambda \cdot \rho(\mathbf{x}, \mathbf{x}'), \quad (3)$$

which can then be solved using gradient-based optimizers if the constraint function ρ is differentiable.

2.1 Text Adversarial Examples

Although the search problem formulation in Equation 2 has been widely applied to continuous data such as image and speech, it does not directly apply to text data because (1) the data space $\mathcal{X}$ is discrete, hence not permitting gradient-based optimization; and (2) the constraint function ρ is difficult to define for text data. In fact, both issues arise when considering attacks against *any* discrete input domain, but the latter is especially relevant for text data due to the sensitivity of natural language. For instance, inserting the word *not* into a sentence can negate the meaning of the whole sentence despite having a token-level edit distance of 1.

Prior work. Several attack algorithms have been proposed to circumvent these two issues, using a multitude of approaches. For attacks that operate on the character level, perceptibility can be approximated by the number of character edits, *i.e.*, replacements, swaps, insertions and deletions (Ebrahimi et al., 2017; Li et al., 2018; Gao et al., 2018). Attacks that operate on the word level adopt heuristics such as synonym substitution (Samanta and Mehta, 2017; Zang et al., 2020; Maheshwary et al., 2020) or replacing words by ones with similar word embeddings (Alzantot et al., 2018; Ren et al., 2019; Jin et al., 2020). More recent attacks have also leveraged masked language models such as BERT (Devlin et al., 2019) to generate word substitutions by replacing masked tokens (Garg and Ramakrishnan, 2020; Li et al., 2020a,b). Most of the aforementioned attacks follow the common recipe of proposing character-level or word-level perturbations to generate a constrained candidate set and optimizing the adversarial loss greedily or using beam search.

Shortcomings in prior work. Despite the plethora of attacks against natural language models, their efficacy remains subpar compared to attacks against other data modalities. Both character-level and word-level changes are still relatively detectable, especially as such changes often introduce misspellings, grammatical errors, and other artifacts of unnaturalness in the perturbed text (Morris et al., 2020a). Moreover, prior attacks mostly query the target model h as a black-box and rely on zeroth-order strategies for minimizing the adversarial loss, resulting in sub-optimal performance.

For instance, BERT-Attack (Li et al., 2020b)—arguably the state-of-the-art attack against BERT—only reduces the test accuracy of the target model on the AG News dataset (Zhang et al., 2015) from 95.1 to 10.6. In comparison, attacks against image models can consistently reduce the model's accuracy to 0 on almost all computer vision tasks (Akhtar and Mian, 2018). This gap in performance raises the question of whether gradient-based search can produce more fluent and optimal adversarial examples on text data. In this work, we show that our gradient-based attack can reduce the same model's accuracy from 95.1 to 3.5 while being *more semantically-faithful* to the original text. Our result shows that using gradient-based search for text adversarial examples can indeed close the performance gap between vision and text attacks.

2.2 Other Attacks

While most works on adversarial attack on text fall within the formulation defined at the beginning of section 2, other notions of adversarial perturbation exist as well. One class of such attacks is known as universal adversarial triggers—a short snippet of text that when appended to any input, causes the model to misclassify (Wallace et al., 2019; Song et al., 2020). However, such triggers often contain unnatural combinations of words or tokens, and hence are very perceptible to a human observer.

Our work falls within the general area of adversarial learning, and many prior works in this area have explored the notion of adversarial example on different data modalities. Although the most prominent data modality by far is image, adversarial examples can be constructed for speech (Carlini and Wagner, 2018) and graphs (Dai et al., 2018; Zügner et al., 2018) as well.

3 GBDA: Gradient-based Distributional Attack

In this section, we detail GBDA—our general-purpose framework for gradient-based text attacks against transformers. Our framework leverages two important insights: (1) we define a parameterized *adversarial distribution* that enables gradient-based search using the Gumbel-softmax (Jang et al., 2016); and (2) we promote fluency and semantic faithfulness of the perturbed text using soft constraints on both perplexity and semantic similarity.

3.1 Adversarial Distribution

Let $\mathbf{z} = z_1 z_2 \cdots z_n$ be a sequence of tokens where each $z_i \in \mathcal{V}$ is a token from a fixed vocabulary

$\mathcal{V} = \{1, \ldots, V\}$. Consider a distribution P_Θ parameterized by $\Theta \in \mathbb{R}^{n \times V}$, which yields samples $\mathbf{z} \sim P_\Theta$ by independently sampling each token $z_i \sim \text{Categorical}(\pi_i)$, where $\pi_i = \text{Softmax}(\Theta_i)$ is a vector of token probabilities for the i-th token.

We aim to optimize the parameter matrix Θ so that samples $\mathbf{z} \sim P_\Theta$ are adversarial examples for the model h. To do so, we define the objective function for this goal as:

$$\min_{\Theta \in \mathbb{R}^{n \times V}} \mathbb{E}_{\mathbf{z} \sim P_\Theta} \ell(\mathbf{z}, y; h), \tag{4}$$

where ℓ is a chosen adversarial loss.

Extension to probability vector inputs. The objective function in Equation 4 is non-differentiable due to the discrete nature of the categorical distribution. Instead, we propose a relaxation of Equation 4 by first extending the model h to take probability vectors as input, and then use the Gumbel-softmax approximation (Jang et al., 2016) of the categorical distribution to derive the gradient.

Transformer models take as input a sequence of tokens that are converted to embedding vectors using a lookup table. Let $\mathbf{e}(\cdot)$ be the embedding function so that the input embedding for the token z_i is $\mathbf{e}(z_i) \in \mathbb{R}^d$ for some embedding dimension d. Given a probability vector π_i that specifies the sampling probability of the token z_i, we define

$$\mathbf{e}(\pi_i) = \sum_{j=1}^{V} (\pi_i)_j \mathbf{e}(j) \tag{5}$$

as the embedding vector corresponding to the probability vector π_i. Note that if π_i is a one-hot vector corresponding to the token z_i then $\mathbf{e}(\pi_i) = \mathbf{e}(z_i)$. We extend the notation for an input sequence of probability vectors $\boldsymbol{\pi} = \pi_1 \cdots \pi_n$ as $\mathbf{e}(\boldsymbol{\pi}) = \mathbf{e}(\pi_1) \cdots \mathbf{e}(\pi_n)$ by concatenating the input embeddings.

Computing gradients using Gumbel-softmax. Extending the model h to take probability vectors as input allows us to leverage the Gumbel-softmax approximation to derive smooth estimates of the gradient of Equation 4. Samples $\tilde{\boldsymbol{\pi}} = \tilde{\pi}_1 \cdots \tilde{\pi}_n$ from the Gumbel-softmax distribution $\tilde{P}_\Theta$ are drawn according to the process:

$$(\tilde{\pi}_i)_j := \frac{\exp((\Theta_{i,j} + g_{i,j})/T)}{\sum_{v=1}^{V} \exp((\Theta_{i,v} + g_{i,v})/T)}, \tag{6}$$

where $g_{i,j} \sim \text{Gumbel}(0, 1)$ and $T > 0$ is a temperature parameter that controls the smoothness of the Gumbel-softmax distribution. As $T \to 0$, this distribution converges towards the distribution $\text{Categorical}(\text{Softmax}(\Theta_i))$.

We can now optimize Θ using gradient descent by defining a smooth approximation of the objective function in Equation 4:

$$\min_{\Theta \in \mathbb{R}^{n \times V}} \mathbb{E}_{\tilde{\boldsymbol{\pi}} \sim \tilde{P}_\Theta} \ell(\mathbf{e}(\tilde{\boldsymbol{\pi}}), y; h), \tag{7}$$

The expectation can be estimated using stochastic samples of $\tilde{\boldsymbol{\pi}} \sim \tilde{P}_\Theta$.

3.2 Soft Constraints

Black-box attacks based on heuristic replacements can only constrain the perturbation by proposing changes that fall within the constraint set, *e.g.*, limiting edit distance, replacing words by ones with high word embedding similarity, etc. In contrast, our adversarial distribution formulation can readily incorporate any differentiable constraint function as a part of the objective. We leverage this advantage to include both fluency and semantic similarity constraints in order to produce more fluent and semantically-faithful adversarial texts.

Fluency constraint with a language model. Causal language models (CLMs) are trained with the objective of next token prediction by maximizing the likelihood given previous tokens. This allows the computation of likelihoods for any sequence of tokens. More specifically, given a CLM g with log-probability outputs, the negative log-likelihood (NLL) of a sequence $\mathbf{z} = z_1 \cdots z_n$ is given autoregressively by:

$$\text{NLL}_g(\mathbf{z}) = -\sum_{i=1}^{n} \log p_g(z_i \mid z_1 \cdots z_{i-1}),$$

where $\log p_g(z_i \mid z_1 \cdots z_{i-1}) = g(z_1 \cdots z_{i-1})_{z_i}$ is the cross-entropy between the delta distribution on token z_i and the predicted token distribution $g(z_1 \cdots z_{i-1})$ for $i = 1, \ldots, n$.

We extend the definition of NLL to the setting where inputs are vectors of token probabilities by:

$$\begin{aligned} \text{NLL}_g(\boldsymbol{\pi}) &:= -\sum_{i=1}^{n} \log p_g(\pi_i \mid \pi_1 \cdots \pi_{i-1}) \\ &= -\sum_{i=1}^{n} \sum_{j=1}^{V} (\pi_i)_j g(\mathbf{e}(\pi_1) \cdots \mathbf{e}(\pi_{i-1}))_j, \end{aligned}$$

with $\log p_g(\pi_i \mid \pi_1 \cdots \pi_{i-1})$ being the cross-entropy between the *next token distribution*

π_i and the predicted next token distribution $g(\mathbf{e}(\pi_1) \cdots \mathbf{e}(\pi_{i-1}))$. This extension coincides with the NLL for a token sequence $\mathbf{x}$ when each π_i is a delta distribution for the token x_i.

Similarity constraint with BERTScore. Prior work on word-level attacks often used context-free embeddings such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) or synonym substitution to constrain semantic similarity between the original and perturbed text (Alzantot et al., 2018; Ren et al., 2019; Jin et al., 2020). These constraints tend to produce out-of-context and unnatural changes that alter the semantic meaning of the perturbed text (Garg and Ramakrishnan, 2020). Instead, we propose to use BERTScore (Zhang et al., 2019), a similarity score for evaluating text generation that captures the semantic similarity between pairwise tokens in contextualized embeddings of a transformer model.

Let $\mathbf{x} = x_1 \cdots x_n$ and $\mathbf{z} = z_1 \cdots z_m$ be two token sequences and let g be a language model that produces contextualized embeddings $\phi(\mathbf{x}) = (\mathbf{u}_1, \ldots, \mathbf{u}_n)$ and $\phi(\mathbf{z}) = (\mathbf{v}_1, \ldots, \mathbf{v}_m)$. The (recall) BERTScore between $\mathbf{x}$ and $\mathbf{z}$ is defined as:

$$R_{\text{BERT}}(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{n} w_i \max_{j=1,\ldots,m} \mathbf{u}_i^\top \mathbf{v}_j, \quad (8)$$

where $w_i := \text{idf}(x_i) / \sum_{i=1}^{n} \text{idf}(x_i)$ is the normalized inverse document frequency of the token x_i computed across a corpus of data. We can readily substitute $\mathbf{z}$ with a sequence of probability vectors $\boldsymbol{\pi} = \pi_1 \cdots \pi_m$ as described in Equation 5 and use $\rho_g(\mathbf{x}, \boldsymbol{\pi}) = 1 - R_{\text{BERT}}(\mathbf{x}, \boldsymbol{\pi})$ as a differentiable soft constraint.

Objective function. We combine all the components in the previous sections into a final objective for gradient-based optimization. Our objective function uses the margin loss (*cf.* Equation 1) as the adversarial loss, and integrates the fluency constraint with a causal language model g and the BERTScore similarity constraint using contextualized embeddings of g:

$$\begin{aligned}\mathcal{L}(\Theta) =& \mathbb{E}_{\tilde{\boldsymbol{\pi}} \sim \tilde{P}_\Theta} \ell(\mathbf{e}(\tilde{\boldsymbol{\pi}}), y; h) \\ &+ \lambda_{\text{lm}} \, \text{NLL}_g(\tilde{\boldsymbol{\pi}}) + \lambda_{\text{sim}} \, \rho_g(\mathbf{x}, \tilde{\boldsymbol{\pi}}), \quad (9)\end{aligned}$$

where $\lambda_{\text{lm}}, \lambda_{\text{sim}} > 0$ are hyperparameters that control the strength of the soft constraints. We minimize $\mathcal{L}(\Theta)$ stochastically using Adam (Kingma and Ba, 2014) by sampling a batch of inputs from $\tilde{P}_\Theta$ at every iteration.

3.3 Sampling Adversarial Texts

Once Θ has been optimized, we can sample from the adversarial distribution P_Θ to construct adversarial examples. Since the loss function $\mathcal{L}(\Theta)$ that we optimize is an approximation of the objective in Equation 4, it is possible that some samples are not adversarial even when $\mathcal{L}(\Theta)$ is successfully minimized. Hence, in practice, we draw multiple samples $\mathbf{z} \sim P_\Theta$ and stop sampling either when the model misclassifies the sample or when we reach a maximum number of samples.

Note that this stage could technically allow us to add hard constraints to the examples we generate, *e.g.*, manually filter out adversarial examples that do not seem natural. In our case, we do not add any extra hard constraint and only verify that the generated example is misclassified by the model.

Transfer to other models. Since drawing from the distribution P_Θ could potentially generate an infinite stream of adversarial examples, we can leverage these generated samples to query a target model that is different from h. This constitutes a black-box *transfer attack* from the source model h. Moreover, our transfer attack does not require the target model to output continuous-valued scores, which most existing black-box attacks against transformers rely on (Jin et al., 2020; Garg and Ramakrishnan, 2020; Li et al., 2020a,b). We demonstrate in subsection 4.2 that this transfer attack enabled by the adversarial distribution P_Θ is very effective at attacking a variety of target models.

4 Experiments

In this section, we empirically validate our attack framework on a benchmark suite of natural language tasks. Code to reproduce our results is open sourced on GitHub[1].

4.1 Setup

Tasks. We evaluate on several benchmark text classification datasets, including **DBPedia** (Zhang et al., 2015) and **AG News** (Zhang et al., 2015) for article/news categorization, **Yelp Reviews** (Zhang et al., 2015) and **IMDB** (Maas et al., 2011) for binary sentiment classification, and **MNLI** (Williams et al., 2017) for natural language inference. The MNLI dataset contains two evaluation sets:

[1] https://github.com/facebookresearch/text-adversarial-attack

Task	GPT-2			XLM (en-de)			BERT		
	Clean Acc.	Adv. Acc.	Cosine Sim.	Clean Acc.	Adv. Acc.	Cosine Sim.	Clean Acc.	Adv. Acc.	Cosine Sim.
DBPedia	99.2	5.2	0.91	99.1	7.6	0.80	99.2	7.1	0.80
AG News	94.8	6.6	0.90	94.4	5.4	0.87	95.1	3.5	0.80
Yelp	97.8	2.9	0.94	96.3	3.4	0.93	97.3	4.4	0.94
IMDB	93.8	7.6	0.98	87.6	0.1	0.97	93.0	1.8	0.96
MNLI (m.)	81.7	2.8/11.0	0.82/0.88	76.9	5.4/13.1	0.84/0.86	84.6	5.5/11.3	0.82/0.86
MNLI (mm.)	82.5	4.2/13.5	0.85/0.88	76.3	4.1/10.6	0.85/0.86	84.5	4.7/11.8	0.80/0.87

Table 1: Result of white-box attack against three transformer models: GPT-2, XLM (en-de), and BERT. For MNLI, the pair of numbers correspond to result for attacking the hypothesis/premise portions of the text. Our attack is able to reduce the target model's accuracy to below 10% in almost all cases, while maintaining a high level of semantic similarity (cosine similarity of higher than 0.8 using USE embeddings).

Task	Clean Acc.	Attack Alg.	Adv. Acc.	# Queries	Cosine Sim.
AG News	95.1	GBDA (ours)	**8.8**	**107**	0.69
		BERT-Attack	10.6	213	0.63
		BAE	13.0	419	**0.75**
		TextFooler	12.6	357	0.57
Yelp	97.3	GBDA (ours)	**2.6**	**43**	0.83
		BERT-Attack	5.1	273	0.77
		BAE	12.0	434	**0.90**
		TextFooler	6.6	743	0.74
IMDB	93.0	GBDA (ours)	**8.5**	**116**	0.92
		BERT-Attack	11.4	454	0.86
		BAE	24.0	592	**0.95**
		TextFooler	13.6	1134	0.86
MNLI (m.)	84.6	GBDA (ours)	**2.3/10.8**	37/133	0.75/0.79
		BERT-Attack	7.9/11.9	**19/44**	0.55/0.68
		BAE	25.4/36.2	68/120	**0.88/0.88**
		TextFooler	9.6/25.3	78/152	0.57/0.65
MNLI (mm.)	84.5	GBDA (ours)	**1.8/13.4**	30/159	0.76/0.80
		BERT-Attack	7/13.7	**24/43**	0.53/0.69
		BAE	19.2/30.3	75/110	**0.88/0.88**
		TextFooler	8.3/22.9	86/162	0.58/0.65

Table 2: Evaluation of black-box model transfer attack from GPT-2 to finetuned BERT classifiers. Unlike the baseline methods, our transfer attack does not require continuous-valued model outputs. See text for details.

matched (m.) and mismatched (mm.), corresponding to whether the test domain is matched or mismatched with the training distribution.

Models. We attack three transformer architectures with our gradient-based white-box attack: GPT-2 (Radford et al., 2019), XLM (Lample and Conneau, 2019) (using the en-de cross-lingual model), and BERT (Devlin et al., 2019). For BERT, we use finetuned models from TextAttack (Morris et al., 2020b) for all tasks except for DBPedia, where finetuned models are unavailable. For BERT on DBPedia and GPT-2/XLM on all tasks, we finetune a pretrained model to serve as the target model.

The soft constraints described in subsection 3.2 utilizes a CLM g with the same tokenizer as the target model. For GPT-2 we use the pre-trained GPT-2 model without finetuning as g, and for XLM we use the checkpoint obtained after finetuning using the CLM objective. For masked language models such as BERT (Devlin et al., 2019), we train a causal language model g on WikiText-103 using the same tokenizer as the target model.

Baselines. We compare against several recent attacks on text transformers: TextFooler (Jin et al., 2020), BAE (Garg and Ramakrishnan, 2020), and BERT-Attack (Li et al., 2020b). All baseline attacks are evaluated on finetuned BERT models from the TextAttack library (Morris et al., 2020b). See subsection 4.2 for details of attack settings.

Hyperparameters. Our adversarial distribution parameter Θ is optimized using Adam (Kingma and Ba, 2014) with a learning rate of 0.3 and a batch size of 10 for 100 iterations. The distribution parameters Θ are initialized to zero except $\Theta_{i,j} = C$ where $x_i = j$ is the i-th token of the clean input. In practice we take $C \in 12, 15$. We use $\lambda_{\text{perp}} = 1$ and cross-validate $\lambda_{\text{sim}} \in [20, 200]$ and $\kappa \in \{3, 5, 10\}$ using held-out data.

4.2 Quantitative Evaluation

White-box attacks. We first evaluate the attack performance under the white-box setting. Table 1 shows the result of our attacks against GPT-2, XLM (en-de), and BERT on different benchmark datasets. Following prior work (Jin et al., 2020), for each task, we randomly select 1000 inputs from the task's test set as attack targets. After optimizing Θ, we draw up to 100 samples $\mathbf{z} \sim P_\Theta$ until the model misclassifies $\mathbf{z}$. The model's accuracy after attack (under the column "Adv. Acc.") is the accuracy evaluated on the last of the drawn samples.

Overall, our attack is able to successfully generate adversarial examples against all three models across the five benchmark datasets. The test accuracy can be reduced to below 10% for almost all models and tasks. Following prior work, we also evaluate the semantic similarity between the adversarial example and the original input using the co-

Target Model	Task	Clean Acc.	Adv. Acc.	# Queries	Cosine Sim.
ALBERT	**AG News**	94.7	7.5	84	0.68
	Yelp	97.5	5.9	76	0.79
	IMDB	93.8	13.1	157	0.87
RoBERTA	**AG News**	94.7	10.7	130	0.67
	IMDB	95.2	17.4	205	0.87
	MNLI (m.)	88.1	4.1/15.1	63/179	0.69/0.76
	MNLI (mm.)	87.8	3.2/15.9	51/189	0.69/0.78
XLNet	**IMDB**	93.8	12.1	149	0.87
	MNLI (m.)	87.2	3.9/13.7	56/162	0.70/0.77
	MNLI (mm.)	86.8	1.7/14.4	32/171	0.70/0.78

Table 3: Result of black-box model transfer attack from GPT-2 to other transformer models. Our attack is achieved by sampling from the same adversarial distribution P_Θ and is able to generalize to the three target transformer models considered in this study.

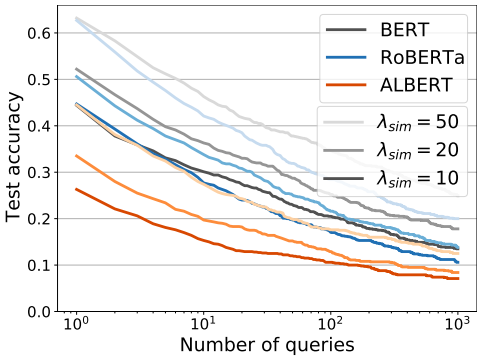

Figure 2: Effect of the parameter λ_{sim} on transfer attack success rate. Lower λ_{sim} produces more aggressive changes, but also generalizes better to different target models.

Task	Architecture	Clean Acc.	Adv. Acc.	# Queries	Cosine Sim.
IMDB → Yelp	**GPT-2**	97.8	22.1	280	0.83
	XLM (en-de)	96.3	7.0	94	0.81
	BERT	97.3	19.3	235	0.79
	GPT-2 → BERT	97.3	26.1	319	0.83
Yelp → IMDB	**GPT-2**	93.8	3.2	52	0.89
	XLM (en-de)	87.6	14.4	163	0.88
	BERT	93.0	15.4	192	0.88
	GPT-2 → BERT	93.0	11.1	138	0.89

Table 4: Evaluation of black-box dataset transfer attack on Yelp/IMDB. Even without access to the target model's data distribution, it is still possible to execute the attack by using GBDA on a model trained for the same task but a different training distribution. See text for details.

sine similarity of Universal Sentence Encoders (Cer et al., 2018) (USE). Our attack is able to consistently maintain a high cosine similarity to the original input (higher than 0.8) in most cases.

Model transfer attacks. We also evaluate our attack against prior work under the black-box setting by transferring across models. More specifically, for each model and task, we randomly select 1000 test samples and optimize the adversarial distribution P_Θ on GPT-2. After optimizing Θ, we draw up to 1000 samples $\mathbf{z} \sim P_\Theta$ and evaluate them on the target BERT model from the TextAttack library (Morris et al., 2020b) until the model misclassifies $\mathbf{z}$. This attack setting is strictly more restrictive than prior work because our query procedure only requires the target model to output a discrete label in order to decide when to stop sampling from P_Θ, whereas prior work relied on a continuous-valued output score such as class probabilities.

Table 2 shows the performance of our attack when transferred to finetuned BERT text classifiers. In all settings, GBDA is able to reduce the target model's accuracy to below that of BERT-Attack and BAE within similar or fewer number of queries. Moreover, the cosine similarity between the original input and the adversarial example is higher than that of BERT-Attack.

We further evaluate our model transfer attack against three other finetuned transformer models from the TextAttack library: ALBERT (Lan et al., 2019), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019). For this experiment, we use the same Θ optimized on GPT-2 for each of the target models. Table 3 reports the performance of our attack after randomly sampling up to 1000 times from P_Θ. The attack performance is comparable to that of the transfer attack against BERT in Table 2, which means our adversarial distribution P_Θ is able to capture the common failure modes of a wide variety of transformer models.

Dataset transfer attacks. The model transfer attack relies on the assumption that the adversary has access to the target model's training data. We relax this assumption in the form of a *dataset transfer attack* where only the target model's task is known. Concretely, we attack sentiment classifiers trained on Yelp/IMDB by using a model trained on one dataset for optimizing Θ and drawing up to 1000 samples from P_Θ to attack the target model trained on the other dataset.

Table 4 shows the result of the dataset transfer attack for different target model architectures. In all except for the case of **GPT-2→BERT**, the model

Attack	Prediction	Text
Original	Entailment (83%)	He found himself thinking in circles of worry and pulled himself back to his problem. He got lost in loops of worry, but snapped himself back to his problem.
GBDA	Neutral (95%)	He found himself thinking in circles of worry and pulled himself back to his problem. He got lost in loops of hell, but snapped himself back to his problem.
Original	Contradiction (78%)	Steps are initiated to allow program board membership to reflect the clienteligible community and include representatives from the funding community, corporations and other partners. There isn't a fair representation of board members on the program.
GBDA	Neutral (98%)	Steps are initiated to allow program board membership to reflect the clienteligible community and include representatives from the funding community, corporations and other partners. There isn also a fair representation of board members on the program..
Original	Contradiction (98%)	Pesticide concentrations should not exceed USEPA's Ambient Water Quality chronic criteria values where available. There is no assigned value for maximum pesticide concentration in water.
GBDA	Entailment (86%)	Pesticide concentrations should not exceed USEPA's Ambient Water Quality chronic criteria values where available. There is varying assigned value for maximum pesticide concentration in water.

Table 5: Examples of successful adversarial texts on the MNLI dataset.

Attack	Prediction	Text
Original	World (99%)	Turkey a step closer to Brussels The European Commission is set to give the green light later today to accession talks with Turkey. EU leaders will take a final decision in December.
GBDA w/ fluency	Business (100%)	Turkey a step closer to Brussels The eurozone Union is set to give the green light later today to accession talks with Barcelona. EU leaders will take a final decision in December.
GBDA w/o fluency	Business (77%)	Turkey a step closer to Uber Thecom Commission is set to give the green light later today to accessrage negotiations with Turkey. EU leaders will take a final decision in December.

Table 6: Examples of adversarial text on AG News generated with and without the fluency constraint. Without the fluency constraint, the constructed adversarial text tends to contain more nonsensical token combinations.

used when optimizing P_Θ has the same architecture as the target model. In the last setting, we simultaneously transfer between the model and the dataset. It is evident that the transfer attack remains successful despite not having access to the target model's training data. This result opens a practical avenue of attack against real world systems as the attacker requires very limited knowledge of the target model in order to succeed.

4.3 Analysis

Sample adversarial texts. Table 5 shows examples of our adversarial attack on text. Our method introduces minimal changes to the text, preserving most of the original sentence's meaning. Despite not explicitly constraining replaced words to have the same Part-Of-Speech tag, we observe that our soft penalties make the adversarial examples obey this constraint. For instance, in the first and third examples of Table 5, "worry" is replaced with "hell" and "no" with "varying".

Effect of λ_{sim}. Figure 2 shows the impact of the similarity constraint on transfer attack adversarial accuracy for GPT-2 on AG News. Each color corresponds to a different target model, whereas the color shade (from light to dark) indicates the value of the constraint hyperparameter: $\lambda_{\text{sim}} = 50, 20, 10$. A higher value of λ_{sim} reduces the aggressiveness of the perturbation, but also increases the number of queries required to achieve a given target adversarial accuracy.

Impact of the fluency constraint. Table 6 shows adversarial examples for GPT-2 on AG News, generated with and without the fluency constraint. We fix all hyperparameters except for the fluency regularization constant λ_{lm}, and sample successful adversarial texts from P_Θ after Θ has been optimized. It is evident that the fluency constraint promotes token combinations to form valid words and ensure grammatical correctness of the adversarial text. Without the fluency constraint, the adversarial text tends to contain nonsensical words.

Tokenization artifacts. Our attack operates entirely on tokenized inputs. However, the input to the classification system is often in raw text form, which is then tokenized before being fed to the model. Thus it is possible that we generate an adversarial example that, when converted to raw text, is not re-tokenized to the same set of tokens.

Consider this example: our adversarial example contains the tokens "jui-" and "cy", which decodes into "juicy", and is then re-encoded to "juic-" and "y". In practice, we observe that these re-tokenization artifacts are rare: the "token error rate" is around 2%. Furthermore, they do not impact adversarial accuracy by much: the re-tokenized example is in fact still adversarial. One potential

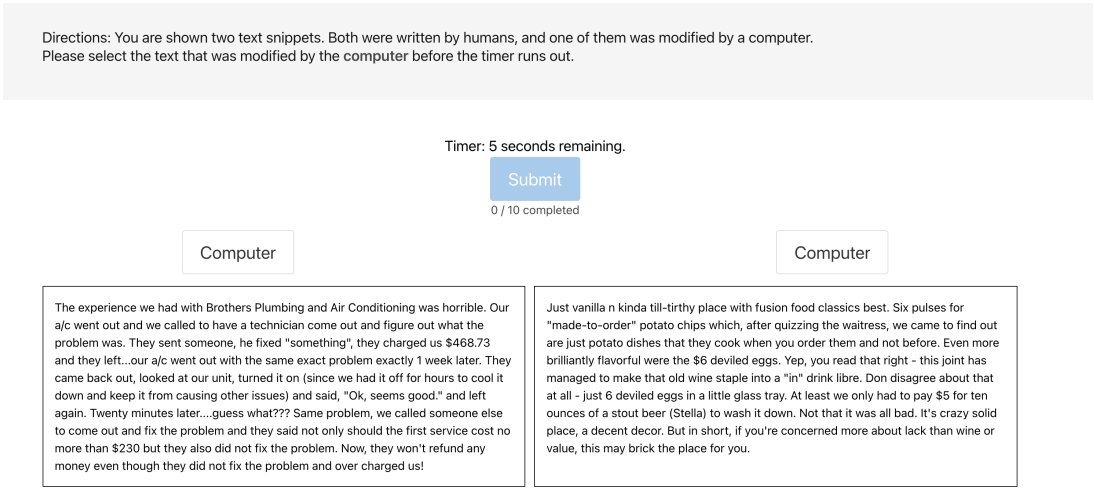

Figure 3: Web interface for the human evaluation experiment using Amazon Mechanical Turk.

mitigation strategy is to re-sample from P_Θ until the sampled text is stable under re-tokenization. Note that all our adversarial accuracy results are computed after re-tokenization.

Runtime. Our method relies on white-box optimization and thus necessitates forward and backward passes through the attacked model, the language model and the similarity model, which increases the per-query time compared to black-box attacks that only compute forward passes. However, this is compensated by a much more efficient optimization which brings the total runtime to $20s$ per generated example, on par with black-box attacks such as BERT-Attack (Li et al., 2020b).

Human evaluation. We further conduct a human evaluation study of our attacks to examine to what extent are adversarial texts generated by GBDA truly imperceptible. Our interface is shown in Figure 3: We show annotators on Amazon Mechanical Turk two snippets of text—one is not modified, and the other one is adversarially corrupted—and the annotator has to select which one is corrupted in less than 10 seconds. The clean text is sampled from Yelp and the adversarial text is generated against BERT using either our method or BAE, our strongest baseline. To ensure high quality of the annotations, we select annotators with more than 1000 hits approved and with an approval rate higher than 98%. The annotation itself is preceded by an onboarding with three simple examples that have to be correctly classified in order for the annotator to qualify for the task.

Averaging across more than 3000 samples, annotators are able to detect BAE examples in 78.04% of the cases, while detecting our examples in 76.06% of the cases. This result shows that although both GBDA and BAE produce detectable changes, our method is slightly less perceptible than BAE but the model accuracy after attack is significantly lower for our attack: 4.7% for GBDA compared to 12.0% for BAE (*cf.* Tables 1 and 2).

5 Conclusion and Future Work

We presented GBDA, a framework for gradient-based white-box attack against text transformers. Our approach overcomes many ad-hoc constraints and limitations from the existing text attack literature by leveraging a novel adversarial distribution formulation, allowing end-to-end optimization of the adversarial loss and fluency constraints with gradient descent. This makes our method generic and potentially applicable to any model for token sequence prediction.

Limitations. One clear limitation of GBDA is its restriction to only token replacements. Indeed, our adversarial distribution formulation using the Gumbel-softmax does not trivially extend to token insertions and deletions. This limitation may adversely affect the naturalness of the generated adversarial examples. We hope to extend our framework to incorporate a broader set of token-level changes in the future.

In addition, the adversarial distribution P_Θ is highly over-parameterized. Despite most adversarial examples requiring only a few token changes, the distribution parameter Θ is of size $n \times V$, which is especially excessive for longer sentences. Future work may be able to reduce the number of parameters without affecting attack performance.

References

Naveed Akhtar and Ajmal Mian. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430.

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*.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer.

Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE.

Nicholas Carlini and David Wagner. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7. IEEE.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. 2018. Adversarial attack on graph structured data. In *International conference on machine learning*, pages 1115–1124. PMLR.

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.

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

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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

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*.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2020a. Contextualized perturbation for textual adversarial attack. *arXiv preprint arXiv:2009.07502*.

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. 2020b. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

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: Human language technologies*, pages 142–150.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Rishabh Maheshwary, Saket Maheshwary, and Vikram Pudi. 2020. Generating natural language attacks in a hard label black box setting. *arXiv preprint arXiv:2012.14956*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*.

John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020a. Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3829–3839, Online. Association for Computational Linguistics.

John X Morris, Eli Lifland, Jin Yong Yoo, and Yanjun Qi. 2020b. Textattack: A framework for adversarial attacks in natural language processing. *arXiv preprint arXiv:2005.05909*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

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

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, Florence, Italy. Association for Computational Linguistics.

Suranjana Samanta and Sameep Mehta. 2017. Towards crafting text adversarial samples. *arXiv preprint arXiv:1707.02812*.

Liwei Song, Xinwei Yu, Hsuan-Tung Peng, and Karthik Narasimhan. 2020. Universal adversarial attacks with natural triggers for text classification. *arXiv preprint arXiv:2005.00174*.

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. *arXiv preprint arXiv:1908.07125*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

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, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *arXiv preprint arXiv:1509.01626*.

Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. 2018. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2847–2856.